(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) Publication number : **0 575 100 A1**

(12) EUROPEAN PATENT APPLICATION

(21) Application number : **93304514.8**

(22) Date of filing : **10.06.93**

(51) Int. Cl.[5] : **G01R 31/02**, H04L 12/26, H04Q 1/14

(30) Priority : **10.06.92 US 896420**
07.09.92 IL 103092

(43) Date of publication of application :
22.12.93 Bulletin 93/51

(84) Designated Contracting States :
AT BE CH DE DK ES FR GB GR IE IT LI LU MC NL PT SE

(71) Applicant : **RIT TECHNOLOGIES LTD.**
7 Ha-arad Street
Tel Aviv 69710 (IL)

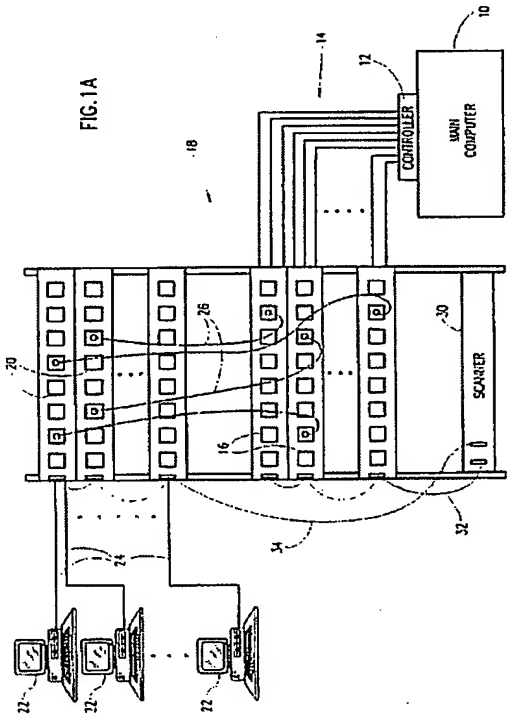
(72) Inventor : **Krupka, Yaakov**
10, Bustanai Street
Rehovot 76280 (IL)
Inventor : **Zisapel, Yehuda**
23 Kessofim Street
Tel Aviv 69355 (IL)

(74) Representative : **Hillier, Peter**
Reginald W. Barker & Co., 13, Charterhouse Square
London, EC1M 6BA (GB)

(54) Patching panel scanner.

(57) A local area network comprising cabling interconnecting a plurality of workstations, the cabling including a plurality of data ports and conductors for selectable and removable interconnection between selected ones of the data ports and apparatus for automatically providing an indication of the connection pattern of the data ports.

FIG.1A



EP 0 575 100 A1

FIELD OF THE INVENTION

The present invention relates to computer communication generally and more particularly to apparatus for interconnection of data ports.

BACKGROUND OF THE INVENTION

Computer communications have become commonplace in recent years. In modern work environments, local area networks are provided for interconnecting a plurality of computers and peripheral apparatus. Generally, the local area networks incorporate computer cabling systems which include distribution panels at which connection between various computer ports and user ports are located.

In many cases, the interconnections between the various ports are relatively complicated and often create a cabling spaghetti, which is extremely difficult to manage. As a result computer software has been developed to enable management of cabling systems. A review of such software appears in "The Great Cabling Treasure Hunt" by M. Jander, Data Communications, March 21, 1991.

Even using the most advanced cable management software, there nevertheless remains a massive job of manually entering connection information for use by the software.

SUMMARY OF THE INVENTION

The present invention seeks to provide apparatus for obviating the present manual task of identifying and collecting cable connection information.

There is thus provided in accordance with a preferred embodiment of the present invention apparatus for providing an indication of the connection pattern of a multiplicity of data ports, pluralities of which are interconnected by conductors, the apparatus including:

signal transducer means operatively associated with at least some of the conductors at the ends thereof adjacent the data ports, at least one of the signal transducer means associated with at least one of the conductors being operative to impose a signal on a portion of the conductor and at least one of the signal transducer means associated with at least one of the conductors being operative to pick off the signal from the conductor;

means, connected to the transducer means, for identifying the existence of signal paths along the conductors between the pluralities of ports; and

output means, coupled to the means for identifying, for providing an output indication of the connection pattern produced by connection of the conductors to the pluralities of ports.

In accordance with a preferred embodiment of the invention, the apparatus also comprises means for distinguishing signals passing along predetermined signal paths from signals passing along other signal paths, according to their phase.

Additionally or alternatively, the apparatus includes means for distinguishing signals passing along predetermined signal paths from signals passing along other signal paths, according to their delay time constant.

Additionally or alternatively, the apparatus includes means for distinguishing signals passing along predetermined signal paths from signals passing along other signal paths, according to their amplitude.

In a preferred embodiment of the the present invention, at least some of the conductors are arranged in a plurality of cables, each cable including at least one conductor.

Further, in accordance with a preferred embodiment of the present invention, the plurality of cables includes shielded cables and the signal transducer means includes induction means operatively associated with shielding of the shielded cables at the ends thereof adjacent the data ports, at least one of the induction means associated with each shielded cable being operative to impose a signal on the shielding of the cable and at least one of the induction means associated with each shielded cable being operative to pick off the signal from the shielding of the cable.

In a preferred embodiment of the invention, the signal transducer means is operative to impose a signal on at least one conductor which does not carry any other signal.

Additionally or alternatively, the signal transducer means is operative to impose a signal on at least one conductor which may carry other signals and includes means for isolating the signal imposed thereby from the other signals, thereby to prevent unacceptable interference therewith.

According to a preferred aspect of this embodiment of the invention the means for isolating includes means for differentiating signals by phase.

Additionally or alternatively, the means for isolating includes means for differentiating signals by delay time constant.

Additionally or alternatively, the means for isolating includes means for differentiating signals by amplitude.

In a preferred embodiment of the invention, the indication of the connection pattern of the data ports is provided automatically.

Additionally, in accordance with a preferred embodiment of the invention, the apparatus includes visual indicators associated with each of the multiplicity of data ports and apparatus for simultaneously operating the visual indicators associated with interconnected data ports, thereby to provide a visible indication of the interconnection therebetween.

In a preferred embodiment of the invention, the visual indicators are LEDs electrically associated with the induction means.

Further, in accordance with a preferred embodiment, the apparatus includes light source apparatus associated with each of the ports for providing a visible indication of pairs of interconnected ports.

In a preferred embodiment of the invention, the apparatus also includes manually controllable means for scanning the ports to provide indication of the pairs of interconnected ports by the light source apparatus.

In accordance with another, preferred, embodiment of the present invention there is provided a local area network including cabling interconnecting a plurality of workstations, the cabling including indication apparatus according to any of the embodiments described above wherein the conductors are used for selectable and removable interconnection between selected ones of the data ports.

In accordance with yet another, preferred, embodiment of the invention there is provided a computer system including at least one main computer, a plurality of workstations and a local area network interconnecting the at least one main computer and the plurality of workstations, the local area network including indication apparatus according to any of the embodiments described above wherein the multiplicity of data ports includes at least one computer port and a plurality of user ports and wherein the conductors are used for selectable and removable interconnection between selected ones of the user ports and the at least one computer port, thereby providing an indication of the connection pattern of the at least one computer port and the user ports.

BRIEF DESCRIPTION OF THE DRAWINGS

The present invention will be more fully understood and appreciated from the following detailed description, taken in conjunction with the drawings in which:

Figs. 1A, 1B and 1C are simplified illustrations of a computer system constructed and operative in accordance with three alternative preferred embodiments of the invention;

Fig. 2 is a simplified detailed illustration of part of the system of Figs. 1A - 1C;

Fig. 3 is a simplified detailed illustration of part of the apparatus of Fig. 2;

Fig. 4 is a simplified detailed illustration of part of the apparatus of Fig. 3;

Fig. 5 is a side view illustration taken along the lines V - V of Fig. 4;

Figs. 6A, 6B, 6C, 6D and 6E are simplified illustrations of five typical circuit arrangements useful in the apparatus of Figs. 3 - 5;

Fig. 7 is a simplified illustration of an alternative embodiment of part of the apparatus of Figs. 1A - 1C and 2;

Fig. 8 is a simplified illustration of a further alternative embodiment of part of the apparatus of Figs. 1A - 1C and 2;

Fig. 9 is a simplified conceptual illustration of the apparatus of Figs. 3 - 6E, indicating an aspect of its operation;

Figs. 10A and 10B are illustrations of signal paths which occur for two different connection arrangements in the apparatus of Fig. 9;

Figs. 11A and 11B are timing diagrams illustrating differentiation between desired signals and interference along the signal paths of Figs. 10A and 10B;

Fig. 12 is an illustration of signal paths which occur for another two alternative connection arrangements in the apparatus of Fig. 9;

Figs. 13A and 13B are timing diagrams illustrating differentiation between desired signals and interference along the signal paths of Fig. 12;

Fig. 14 is an overall diagram of a specific embodiment of the scanner apparatus of Fig. 2;

Figs. 15A, 15B and 15C are simplified illustrations of a computer system constructed and operative in accordance with three additional alternative preferred embodiments of the invention;

Fig. 16 is a simplified detailed illustration of part of the apparatus of Fig. 3 as modified in accordance with the embodiments of Figs. 15A, 15B or 15C; and

Fig. 17 is a simplified illustration of an alternative embodiment of part of the apparatus of Figs. 15A - 15C.

DETAILED DESCRIPTION OF A PREFERRED EMBODIMENT

Reference is now made to Fig. 1A, which illustrates a computer system constructed and operative in accordance with a preferred embodiment of the present invention. The computer system comprises a main computer 10, such as an IBM mainframe, which is coupled, typically via a controller 12, such as a IBM 3270, and local area network cabling 14, typically type I, ICS cabling, to a plurality of computer ports 16 forming part of a distribution panel 18.

The distribution panel 18 also comprises a plurality of user ports 20 to which are coupled individual computers, terminals or work stations 22, hereinafter referred to collectively as "work stations", by apparatus of local area network cabling 24, such as type I, ICS cabling.

Interconnections between individual computer ports 16 and individual user ports 20 are provided by patching cables 26 or alternatively by apparatus of internal connections in patching panels such as the CLPP cordless patching panel commercially available from RIT Technologies Ltd. of Tel Aviv, Israel, and as described in Israel Patent Application 97,227.

In accordance with a preferred embodiment of the present invention, a scanner 30 is provided for automatically and, preferably, repeatedly scanning the interconnection arrangement of the patching cables and, thus, determining the interconnection status of the various computer ports 16 and user ports 20. The scanner 30 may provide an interconnection status output to computer 10 or to any other computer or display in the system, including, for example, a dedicated output device.

Scanner 30 is coupled to computer ports 16 via cables 32 and is coupled to user ports 22 by means of cables 34.

Reference is now made to Fig. 1B, which illustrates a computer system constructed and operative in accordance with another preferred embodiment of the present invention. Similarly to the embodiment of Fig. 1A, the computer system comprises a main computer 10, which is coupled, typically via a controller 12, and local area network cabling 14, to a plurality of computer ports 16 forming part of a distribution panel 18.

As in the embodiment of Fig. 1A, the distribution panel 18 also comprises a plurality of user ports 20 to which are coupled work stations, by means of local area network cabling 24.

As distinguished from the embodiment of Fig. 1A, the distribution panel 18 also includes a plurality of token ring LAN ports 35 which are interconnected by a conventional token ring LAN. Work stations 22 as well as peripheral devices such as printers 36 and disk storage devices 38 may also be coupled to the token ring LAN ports 35.

Interconnections between individual computer ports 16, individual user ports 20 and LAN ports 35 are provided by patching cables 26 or alternatively by means of internal connections in patching panels, as mentioned above.

In accordance with a preferred embodiment of the present invention, scanner 30 is provided for automatically and, preferably, repeatedly scanning the interconnection arrangement of the patching cables and, thus, determining the interconnection status of the various computer ports 16, user ports 20, and LAN ports 35. The scanner 30 may provide an interconnection status output to computer 10 or to any other computer or display in the system, including, for example, a dedicated output device.

Scanner 30 is coupled to computer ports 16 via cables 32, to user ports 22 by means of cables 34 and to LAN ports 35 by means of cables 39.

Reference is now made to Fig. 1C, which illustrates a computer system constructed and operative in accordance with yet another preferred embodiment of the present invention. As distinguished from the embodiments of Figs. 1A and 1B, the computer system need not comprise a main computer but rather may comprise workstations and peripherals, collectively denoted by reference numeral 37, as desired. A multiplicity of ports 40, which need not be classified as computer ports, user ports or LAN ports, are preferably connected to the workstations and peripherals, as the case may be, by local area network cabling 41.

Interconnections between individual ports 40 are provided by patching cables 26 or alternatively by means of internal connections in patching panels, as mentioned above.

In accordance with a preferred embodiment of the present invention, scanner 30 is provided for automatically and, preferably, repeatedly, scanning the interconnection arrangement at least of patching cables 26 and possibly of the internal connections and, thus, determining the interconnection status of the various ports 40. The scanner 30 may provide an interconnection status output to any computer, workstation, or peripheral in the system, including, for example, a dedicated output device.

Scanner 30 is coupled to the various ports via cables 43.

Reference is now made to Fig. 2 which is a more detailed illustration of part of the scanner apparatus of Fig. 1A. It is to be appreciated that the structure illustrated in Fig. 2 and the remaining figures herein is applicable in principle also to the embodiments of Figs. 1A - 1C.

As seen in Fig. 2, the controller 12 is connected via cabling 14 to computer ports 16, which are in turn coupled to scanner 30. Scanner 30 includes a plurality of receiver cards 42, each of which includes a plurality of pre-amplifiers 44, such as OP 16 amplifiers, commercially available from Analog Devices, each inputted from a single port 16 and outputting to a detection matrix 48, whose function is to determine at which user port 20 there are present signals transmitted via a patch cable 26 from a given computer port 16.

A plurality of work stations are connected by means of cabling 24 to corresponding user ports 20 which are coupled, in turn, to driver cards 54 of scanner 30. The driver card 54 includes transmitter logic circuitry 56, a preferred embodiment of which is described hereinafter in greater detail, which outputs to the individual user ports 20 via a plurality of corresponding drivers 58, such as 2N3906 transistors, commercially available from Motorola.

The transmitter logic circuitry 56 receives scanning inputs from a microprocessor 60 which provides overall control functions for the scanner 30 and controls both the detection matrix 48 and the transmitter logic circuitry 56.

Generally speaking, microprocessor 60 causes signal inputs to be applied to the various user ports 20 and scans the computer ports 16 to detect such signal inputs. In such a way, scanner 30 is operative to determine which user ports 20 are connected to which computer ports 16.

Microprocessor 60 is also operative to report information regarding which user ports 20 are connected to which computer ports 16, via a communication interface card 62, either to an adjacent computer 64 or, alternatively, via a pair of modems 66 and another communication interface card 68 to a remotely located computer 70. Alternatively or additionally, the microprocessor 60 may report the connection status via communication interface card 62 to the main computer. The connection to the main computer may be direct or via the LAN system.

It is to be appreciated that the user ports 20 and the computer ports 16 may be interchanged in the system of Fig. 2, such that the driver cards 54 are coupled to the computer ports 16 and the receiver cards 42 are coupled to the user ports 20. It is further noted that, as in the embodiment of Fig. 1C, the distinction between user ports and computer ports may be eliminated entirely, and signals may be transmitted at any selected port and picked off at any selected port. Furthermore, it is envisioned that the interconnection status of ports which are not interconnected by patch cables 26, but rather by other conductors, may also be determined on a real time basis in accordance with the present invention.

Reference is now made to Fig. 3, which illustrates in greater detail the interconnections between an individual user port 20 and an individual computer port 16. The patch cable 26 is shown with connectors 70 and 72 adapted for connection to the user port 20 and the computer port 16.

In accordance with a preferred embodiment of the present invention, there is associated with each of computer ports 16 and user ports 20 an inductive coupler 75, typically comprising a ferromagnetic frame 76, which is wound with a coil 78. The coil 78 of the coupler 75 associated with user port 20 is coupled to the output of driver 58, while the coil 78 of the coupler 75 associated with computer port 16 is coupled to the input of amplifier 44. When patch cable 26 is connected between user port 20 and computer port 16, signals generated from driver 58 are induced by coil 78 of the coupler 75 associated with port 20 onto connector 70 of patch cable 26. The signals carried by patch cable 26 are then induced, by connector 72 at the other end of the cable, onto coil 78 of the coupler 75 associated with port 16 and, thereby, received by amplifier 44. Thus, when port 20 and port 16 are coupled by patch cable 26, as described above, any signal generated from driver 58 is picked up by amplifier 44.

Figs. 4 and 5 illustrate in even greater detail a preferred embodiment of part of a distribution panel 18 including a plurality of ports 16 or 20 together with their inductive couplers 75 and the wiring thereto including matrix diodes 80, such as IN4148 diodes. It is seen that frames 76 typically include a plurality of parallel plates 82 and typically two coils 78 are associated with each frame 76.

Reference is now made to Figs. 6A - 6E, which illustrate three alternative arrangements of couplers 75. The arrangement of Fig. 6A corresponds to that shown in Figs. 4 and 5. The arrangement of Fig. 6B corresponds to that shown in Fig. 3 with the addition of diodes 80. The arrangement of Fig. 6C corresponds to that shown in Fig. 3. Here the diodes 80 are incorporated in the driver card 54 or receiver card 42 (both shown in Fig. 2). Figs. 6D and 6E illustrate arrangements corresponding to those of Figs. 6A and 6B but with the opposite diode direction.

Reference is now made to Fig. 7, which illustrates an alternative embodiment of the invention of Figs. 1 and 2, wherein the inductive couplers 75 are replaced by dry contacts to a conductor coupled to either of the computer port 16 or the user port 20. In Fig. 7, an arrangement is shown wherein the dry contact between the driver card or the receiver card and the port is made to a conductor which is dedicated to this purpose and does not carry any other signal.

Fig. 8 illustrates a variation of the embodiment of Fig. 7, wherein a dry contact connection is made to con-

ductors which also serves to carry other signals. In the embodiment of Fig. 8, a preferred common mode signal imposition technique is employed wherein differential drivers 92 are employed to impose low-frequency signals onto two conductors which also carry data. Differential amplifiers 94 are employed for receiving the imposed signals, preferably through low pass filters which can be seen associated with the signal pick-ups.

By virtue of the low pass filters, only common mode imposed signals are admitted to amplifiers 9. Computer or workstation data signals, preferably having relatively high frequencies, are filtered out by the low pass filters. As can be seen in Fig. 8, high pass filters 90 are interposed between the ports and the main computer or workstations. By virtue of the high pass filters, only data signal are received by the computer or workstations, while the lower frequency common mode signals are filtered out.

Reference is now made to Fig. 9, which is employed to illustrate particular features of the operation of the apparatus of the invention. The patch cable interconnection between ports 16 and 20 provides a signal path l1, such that if a square wave signal, V_{in}, indicated by reference numeral 100 is transmitted via an induction coupling coil 102 along path l1, the time derivative of that signal as received at port 16 and picked up by an induction coupling coil 104 will appear as indicated at reference numeral 106.

Due to unwanted current flows within the distribution panel, there are also present additional, unwanted, signal paths, which are represented in Fig. 9 by a signal path al1, where "a", represents an attenuation factor, less than unity. It has been appreciated by the inventors that the time derivative of the signal 100 which is received over the unwanted signal paths has an opposite polarity from that of the signal 106 received at the opposite end of the patch cable. Accordingly, the signal picked up by coils 108 lying along the unwanted signal path or paths al1 and downstream of a patch cable 26 have a time derivative which appears as indicated by reference numeral 110.

It is a particular feature of the present invention that the polarity difference in the signal time derivative is employed to enable the scanner 30 to differentiate signals passed along a patch cable or its equivalent conductor from signals passed along unwanted signal paths.

Reference is now made to Figs. 10A and 10B which illustrate another aspect of the operation of the apparatus of the invention. Where ports which receive signals from scanner 30 are physically adjacent ports at which signals are picked up, signal flux leakage can result in unwanted signal reception at ports which are not connected by a patch cable to a port receiving that signal.

Fig. 10A illustrates first and second adjacent ports which are not connected by a patch cable or equivalent but which are coupled in an unwanted manner by signal flux leakage. Fig. 10B illustrates similar adjacent ports which are connected by a cable such as patch cable 26 or its equivalent.

Assuming that in both cases a voltage signal of the type illustrated in Fig. 11A is applied to respective induction coils 120 and 122, the signal received at a coil 124 along the patch cable 26 or its equivalent will appear as indicated by solid line V_o and has a first decay time constant, while the signal received at a coil 126 due to signal flux leakage will appear as in in by line aV_o and has a second decay time constant, which is shorter than the first decay time constant. This is true because the ratio of inductance to resistance, which determines the decay time constant, is much greater for the signal path including a patch cable than for a signal path not including a patch cable.

The relationship between the two signals V_o and aV_o is illustrated in Fig. 11B, from where it can be seen that application of positive and/or negative thresholds can be employed to distinguish between the two signals. As can be seen in Fig. 11B, signal aV_o reaches a preselected positive threshold at time t_1, following a positive peak, and a preselected negative threshold at time t_3, following a negative peak. Signal V_o, in contrast, reaches the positive threshold at time t_2, later than t_1, and the negative threshold at time t_4, later than t_3. The differences in times, between t_2 and t_1 and between t_3 and t_4, are useful for distinguishing between signals V_o and aV_o. Circuitry which employs either or both of these thresholds is preferably incorporated into the apparatus of the present invention.

Reference is now made to Fig. 12, 13A and 13B, which are employed to illustrate additional particular features of the operation of the apparatus of the invention. The patch cable interconnection between ports 16 and 20 provides a signal path l1, such that if a square wave signal, V_{in}, indicated by reference numeral 101 in Fig. 13A is transmitted via an induction coupling coil 142 along path l1, the time derivative of that signal as received at port 16 and picked up by an induction coupling coil 144 will appear as indicated at reference numeral 106 in Fig. 13B.

Due to unwanted current flows within the distribution panel, there are, also present additional, unwanted, signal paths, which are represented in Fig. 12 by a signal path bl1, where b, represents an attenuation factor, less than unity. It has been appreciated by the inventors that the time derivative of the signal 101 which is received over the unwanted signal paths has the same polarity as that of the signal 106 received at the opposite end of the patch cable.

Accordingly, the signal bV_o picked up by coils 148 lying along the unwanted signal path or paths bl1 and

downstream of another patch cable 26' is not distinguishable from signal 106 by its time derivative. Furthermore, in contrast to signal aV_o of Fig. 10A, signal bV_o is generally not distinguishable from signal 106 by its delay time constant. However, it has been appreciated by the present inventors that the signals picked up by coils 144 (the wanted signal) and 148 (the unwanted signal) in the arrangement of Fig. 12 may be distinguished by their relative amplitude.

It is a particular feature of an embodiment of the present invention that the amplitude of the signals is employed to enable the scanner 30 to differentiate signals passed along a patch cable or its equivalent conductor directly between connected ports from signals passed along the unwanted signal paths.

The relationship between the two signals V_o and bV_o is illustrated in Fig. 13B from where it can be seen that application of positive and/or negative thresholds can be employed to distinguish between the two signals, as described above with reference to Fig. 11B. As can be seen in Fig. 13B, signal bV_o reaches a preselected positive threshold at time t_1', following a positive peak, and a preselected negative threshold at time t_3', following a negative peak. Signal V_o reaches the positive threshold at time t_2', later than t_1', and the negative threshold at time t_4', later than t_3'. The time differences, namely, t_2' minus t_1' t_3' minus t_4', which result from the different amplitudes of signals V_o and bV_o, are useful for distinguishing between signals V_o and bV_o. Circuitry which employs these thresholds is incorporated into the apparatus of the present invention.

Reference is now made to Fig. 14, which is an overall schematic illustration of a preferred embodiment of the invention corresponding to Fig. 2. Annex I is a net list setting forth with particularity the configuration of a preferred embodiment of the apparatus of Fig. 2. Annex II is a listing of the software embodied in microprocessor 60 (Fig. 2) forming part of the controller of Fig. 14, and which is used to operate the apparatus specified in Annex I.

Reference is now made to Figs. 15A, 15B, 15C, 16 and 17 which illustrate an alternative embodiment of the apparatus of Figs. 1A - 1C, 4 and 7. The apparatus shown in Figs. 15A - 15C, 16 and 17 is similar or identical to that of corresponding Figs. 1A - 1C, 4 and 7 with the addition of a further feature, the provision of a visual indication of interconnection between interconnected data ports by means of light sources associated with the data ports.

As illustrated in Figs. 15A - 15C LEDs 160 are associated with each of the ports 16 and 20. The LEDs 160 are each electrically connected to the circuit which passes through each port. Scanning of each of the ports 16 and 20 to illuminate the LED 160 corresponding to that port and the port connected thereto by patch cord 34 or otherwise may be controlled by a hand operated device 162, such as a joystick, a four directional switch or a mouse. A button 164 can be used to permit scanning to be temporarily stopped.

Additionally or alternatively, the ports 160 may be scanned automatically by scanner 30. In this case, in contrast to normal scanning operation of scanner 30 being that the duration during which the LEDs are illuminated to indicate each pair of interconnected ports 16 and 20 is extended beyond the usual dwell time of the scanner 30 on each port pair. This change is effected by a straightforward change in the software of the scanner 30.

Figs. 16 and 17 illustrate a preferred connection of each LED 160 in series with a resistor 166 between coil 78 and resistor 80 and ground.

It will be appreciated by persons skilled in the art that the present invention is not limited by what has been particularly shown and described hereinabove. Rather the scope of the present invention is defined only by the claims which follow:

ANNEX I

NET LIST FILES
README DOC

*.net - Text files of netlist of controller,receiver and transmitter
*.crf - Text files of cross reference (devices names and values)
*.sch - Schematics files of orcad

THE NETLIST AND CROSS REFERENCE FILES

CONTROL . NET	CONTROL . CRF
TRANSMIT. NET	TRANSMIT. CRF
RCV . NET	RCV. CRF

CONTROLLER NETLIST
FILE: CONTRO1.NET
P.1 - P.4

```
0001 WR0#/1              U1-4  U9-17
0002 DATA/1              U1-3  U2-3  U9-16  U3-3
0003 N00003              U2-9  U5-3
0004 N00004              U2-10  U5-5
0005 N00005              U2-11  U5-7
0006 N00006              U2-12  U5-9
0007 WR1#/1              U2-4  U9-18
0008 N00008              U2-13  U5-11
0009 N00009              U2-14  U5-14
0010 N00010              U2-15  U6-3
0011 N00011              U2-1  U6-5
0012 N00012              R5-2  SW1-2  U9-1
0013 WR2#/1              U9-19  U3-4
0014 N00014              U3-9  U6-7
0015 WREN/1              U9-28  U4-1
0016 N00016              U3-10  U6-9
0017 N00017              C1-1  X1-2  R4-1  U9-39
0018 WRO/1               U9-27  U7-11
0019 N00019              U3-11  U6-11
0020 SCAN#/1             U9-26  U7-14
0021 N00021              U3-12  U6-14
0022 N00022              U3-13  U7-3
0023 N00023              U3-14  U7-5
0024 N00024              U3-15  U7-7
0025 N00025              C2-1  X1-1  R4-2  U9-38
0026 N00026              U3-1  U7-9
0027 RXDC#/1             U9-21  U10-6
0028 A3/1                U4-22  U9-15
0029 RL0/1               U8-2  U9-11
0030 RL1/1               U8-4  U9-10
0031 RL2/1               U8-6  U9-9
0032 RL3/1               U8-8  U9-8
0033 RL4/1               U8-10  U9-7
0034 RL5/1               U8-12  U9-6
0035 RL6/1               U10-2  U9-5
0036 RL7/1               U10-4  U9-4
0037 N00037              RPACK2-1  SW2-16  U11-1
0038 N00038              RPACK2-2  SW2-15  U11-2
0039 N00039              RPACK2-3  SW2-14  U11-3
0040 N00040              RPACK2-4  SW2-13  U11-4
0041 N00041              RPACK2-5  SW2-12  U11-5
0042 N00042              RPACK2-6  SW2-11  U11-6
0043 N00043              RPACK2-7  SW2-10  U11-7
0044 N00044              RPACK2-8  SW2-9  U11-9
0045 N00045              R14-2  Q1-BASE
0046 N00046              R15-2  Q2-BASE
0047 N00047              R16-2  Q3-BASE
0048 N00048              R17-2  Q4-BASE
0049 N00049              R18-2  Q5-BASE
0050 N00050              R19-2  Q6-BASE
0051 N00051              R20-2  Q7-BASE
0052 N00052              R21-2  Q8-BASE
0053 N00053              R51-2  Q9-BASE
0054 N00054              Q9-EMITTER  Q10-EMITTER  Q11-EMITTER  Q12-EMITTER
                         Q13-EMITTER  Q14-EMITTER  R41-1  R40-1
```

1

```
                                R42-1  Q15-EMITTER  Q16-EMITTER
0055 N00055                     R50-2  Q10-BASE
0056 N00056                     R49-2  Q11-BASE
0057 N00057                     R48-2  Q12-BASE
0058 N00058                     C12-1  R43-2  U13-4  R57-1
0059 N00059                     R41-2  U13-5  R54-1
0060 N00060                     R47-2  Q13-BASE
0061 N00061                     R57-2  C11-1  U13-6  R56-1
0062 N00062                     R46-2  Q14-BASE
0063 N00063                     R42-2  R55-1  U13-7
0064 N00064                     R45-2  Q15-BASE
0065 N00065                     R44-2  Q16-BASE
0066 N00066                     R30-2  U12-5  U12-7  R31-1
0067 N00067                     R28-2  U12-2
0068 N00068                     R24-2  U12-1
0069 N00069                     R22-2  R58-1  D2-ANODE  D1-CATHODE
                                U10-11
0070 N00070                     D4-ANODE  D5-ANODE  R23-1
0071 N00071                     R23-2  D3-CATHODE  R26-2  U10-13
0072 IDO/1                      U9-30  R59-2  U12-6
0073 N00073                     P1-2  R25-2  R24-1
0074 RDI/1                      U9-29  U10-10
0075 N00075                     P1-3  R22-1
0076 DTR#/1                     U9-23  U10-12
0077 N00077                     P1-8  P1-6  R29-2  R28-1
0078 N00078                     P1-4  R27-2  D4-CATHODE
0079 DSR#/1                     U9-24  U12-4
0080 TOVF/1                     U9-33  U13-2  R54-2  R52-2
0081 ITF/1                      U9-34  U13-1  R55-2  R53-2
0082 IID/1                      U9-36  U11-14
0083 YDRV0/1                    U5-2  R51-1
0084 YDRV1/1                    U5-4  R50-1
0085 YDRV2/1                    U5-6  R49-1
0086 YDRV3/1                    U5-10  R48-1
0087 YDRV4/1                    U5-12  R47-1
0088 YDRV5/1                    U5-15  R46-1
0089 YDRV6/1                    U6-2  R45-1
0090 YDRV7/1                    U6-4  R44-1
0091 YRCV0/1                    U6-6  R14-1
0092 YRCV1/1                    U6-10  R15-1
0093 YRCV2/1                    U6-12  R16-1
0094 YRCV3/1                    U6-15  R17-1
0095 YRCV4/1                    U7-2  R18-1
0096 YRCV5/1                    U7-4  R19-1
0097 YRCV6/1                    U7-6  R20-1
0098 YRCV7/1                    U7-10  R21-1
0099 A0/1                       U4-2  U3-5  U9-12  U2-5
                                U1-5  U11-11
0100 A1/1                       U4-3  U3-6  U9-13  U2-6
                                U1-6  U11-12
0101 A2/1                       U4-21  U3-7  U9-14  U2-7
                                U1-7  U11-13
0102 N00102                     D5-CATHODE
0103 +VH                        R13-1  R12-1  R11-1  R10-1
                                R9-1  R8-1  R7-1  R6-1
```

		R39-1 R38-1 R37-1 R36-1
		R35-1 R34-1 R33-1 R32-1
		U12-3
0104	-VH	R27-1 U12-12
0105	TEN	U1-1
0106	WRO#	U7-12
0107	SCAN	U7-15
0108	VSSS	Q8-EMITTER C10-2 Q7-EMITTER C9-2
		Q6-EMITTER C8-2 Q5-EMITTER C7-2
		Q4-EMITTER C6-2 Q3-EMITTER C5-2
		Q2-EMITTER C4-2 Q1-EMITTER C3-2
0109	RL#0	RPACK1-16 U8-1
0110	RL#1	RPACK1-15 U8-3
0111	RL#2	RPACK1-14 U8-5
0112	RL#3	RPACK1-13 U8-9
0113	RL#4	RPACK1-12 U8-11
0114	RL#5	RPACK1-11 U8-13
0115	RL#6	RPACK1-10 U10-1
0116	RL#7	RPACK1-9 U10-3
0117	RXDC	R3-2 U10-5
0118	IDT0	R1-2 U9-31
0119	IDT1	R2-2 U9-32
0120	VCC	U10-14 U8-14 U4-24 U7-1
		U3-16 U9-37 U9-2 U6-1
		U9-40 R5-1 U5-1 U2-16
		U1-16 U11-16 SW2-1 SW2-2
		SW2-3 SW2-4 SW2-5 SW2-6
		SW2-7 SW2-8 R53-1 U13-3
		R52-1 R43-1 R59-1 R26-1
		D2-CATHODE R25-1 R29-1 R30-1
0121	GND	U10-7 U10-9 R1-1 R2-1
		U8-7 U4-12 R3-1 U4-23
		U9-20 U7-8 RPACK1-1 RPACK1-2
		RPACK1-3 RPACK1-4 RPACK1-5 RPACK1-6
		RPACK1-7 RPACK1-8 U3-8 C2-2
		U3-2 U6-8 C1-2 SW1-1
		U2-8 U5-8 U2-2 U1-8
		P1-5 U1-2 U11-8 U11-15
		U11-10 RPACK2-16 RPACK2-15 RPACK2-14
		RPACK2-13 RPACK2-12 RPACK2-11 RPACK2-10
		RPACK2-9 R56-2 R40-2 C11-2
		C12-2 U13-12 D3-ANODE R58-2
		D1-ANODE R31-2
0122	TA0	U1-9
0123	TA1	U1-10
0124	TA2	U1-11
0125	TA3	U1-12
0126	TA4	U1-13
0127	TA5	U1-14
0128	TA6	U1-15
0129	AYDRV0	R32-2 Q9-COLLECTOR
0130	AYDRV1	R33-2 Q10-COLLECTOR
0131	AYDRV2	R34-2 Q11-COLLECTOR
0132	AYDRV3	R35-2 Q12-COLLECTOR
0133	AYDRV4	R36-2 Q13-COLLECTOR

0134 AYDRV5	R37-2	Q14-COLLECTOR
0135 AYDRV6	R38-2	Q15-COLLECTOR
0136 AYDRV7	R39-2	Q16-COLLECTOR
0137 AYRCV0	R6-2	Q1-COLLECTOR C3-1
0138 AYRCV1	R7-2	Q2-COLLECTOR C4-1
0139 AYRCV2	R8-2	Q3-COLLECTOR C5-1
0140 AYRCV3	R9-2	Q4-COLLECTOR C6-1
0141 AYRCV4	R10-2	Q5-COLLECTOR C7-1
0142 AYRCV5	R11-2	Q6-COLLECTOR C8-1
0143 AYRCV6	R12-2	Q7-COLLECTOR C9-1
0144 AYRCV7	R13-2	Q8-COLLECTOR C10-1
0145 REN#0	U4-11	
0146 REN#1	U4-9	
0147 REN#2	U4-10	
0148 REN#3	U4-8	
0149 REN#4	U4-7	
0150 REN#5	U4-6	
0151 REN#6	U4-5	
0152 REN#7	U4-4	
0153 REN#8	U4-18	
0154 REN#9	U4-17	
0155 REN#10	U4-20	
0156 REN#11	U4-19	
0157 REN#12	U4-14	
0158 REN#13	U4-13	
0159 REN#14	U4-16	
0160 REN#15	U4-15	

CONTROLLER NETLIST - CROSS REF. FILE CONTROL CRF PARTS 1 TO 120

SCANNER CONTROLER — Revised: April 19, 1992
RIT - CONTROL.SCH — Revision: 1.0
Part Cross Reference Listing — April 19, 1992 20:04:44 — Page 1

Item	Reference	Part	Sheetname	Sheet#	Filename
1	C1	20pF	<<<root>>>	1	CONTROL.SCH
2	C2	20pF	<<<root>>>	1	CONTROL.SCH
3	C3	10nF	YRCVSELECT	3	YRCVSEL.SCH
4	C4	10nF	YRCVSELECT	3	YRCVSEL.SCH
5	C5	10nF	YRCVSELECT	3	YRCVSEL.SCH
6	C6	10nF	YRCVSELECT	3	YRCVSEL.SCH
7	C7	10nF	YRCVSELECT	3	YRCVSEL.SCH
8	C8	10nF	YRCVSELECT	3	YRCVSEL.SCH
9	C9	10nF	YRCVSELECT	3	YRCVSEL.SCH
10	C10	10nF	YRCVSELECT	3	YRCVSEL.SCH
11	C11	0.1uF	YDRIVER	4	YDRIVER.SCH
12	C12	0.1uF	YDRIVER	4	YDRIVER.SCH
13	D1	1N4150	RS232	5	RS232.SCH
14	D2	1N4150	RS232	5	RS232.SCH
15	D3	1N4150	RS232	5	RS232.SCH
16	D4	1N4150	RS232	5	RS232.SCH
17	D5	1N4150	RS232	5	RS232.SCH
18	P1	CONNECTOR DB9	<<<root>>>	1	CONTROL.SCH
19	Q1	2N4401	YRCVSELECT	3	YRCVSEL.SCH
20	Q2	2N4401	YRCVSELECT	3	YRCVSEL.SCH
21	Q3	2N4401	YRCVSELECT	3	YRCVSEL.SCH
22	Q4	2N4401	YRCVSELECT	3	YRCVSEL.SCH
23	Q5	2N4401	YRCVSELECT	3	YRCVSEL.SCH
24	Q6	2N4401	YRCVSELECT	3	YRCVSEL.SCH
25	Q7	2N4401	YRCVSELECT	3	YRCVSEL.SCH
26	Q8	2N4401	YRCVSELECT	3	YRCVSEL.SCH
27	Q9	2N4401	YDRIVER	4	YDRIVER.SCH
28	Q10	2N4401	YDRIVER	4	YDRIVER.SCH
29	Q11	2N4401	YDRIVER	4	YDRIVER.SCH
30	Q12	2N4401	YDRIVER	4	YDRIVER.SCH
31	Q13	2N4401	YDRIVER	4	YDRIVER.SCH
32	Q14	2N4401	YDRIVER	4	YDRIVER.SCH
33	Q15	2N4401	YDRIVER	4	YDRIVER.SCH
34	Q16	2N4401	YDRIVER	4	YDRIVER.SCH
35	RPACK1	22K	<<<root>>>	1	CONTROL.SCH
36	RPACK2	22K	ID-UNIT	2	IDUNIT.SCH
37	R1	22K	<<<root>>>	1	CONTROL.SCH
38	R2	22K	<<<root>>>	1	CONTROL.SCH
39	R3	47K	<<<root>>>	1	CONTROL.SCH
40	R4	10M	<<<root>>>	1	CONTROL.SCH
41	R5	22K	<<<root>>>	1	CONTROL.SCH
42	R6	10K	YRCVSELECT	3	YRCVSEL.SCH
43	R7	10K	YRCVSELECT	3	YRCVSEL.SCH
44	R8	10K'	YRCVSELECT	3	YRCVSEL.SCH
45	R9	10K	YRCVSELECT	3	YRCVSEL.SCH
46	R10	10K	YRCVSELECT	3	YRCVSEL.SCH
47	R11	10K	YRCVSELECT	3	YRCVSEL.SCH
48	R12	10K	YRCVSELECT	3	YRCVSEL.SCH

1

49	R13	10K	YRCVSELECT	3	YRCVSEL.SCH
50	R14	100	YRCVSELECT	3	YRCVSEL.SCH
51	R15	100	YRCVSELECT	3	YRCVSEL.SCH
52	R16	100	YRCVSELECT	3	YRCVSEL.SCH
53	R17	100	YRCVSELECT	3	YRCVSEL.SCH
54	R18	100	YRCVSELECT	3	YRCVSEL.SCH

SCANNER CONTROLER — Revised: April 19, 1992
RIT - CONTROL.SCH — Revision: 1.0
Part Cross Reference Listing — April 19, 1992 20:04:44 — Page 2

Item	Reference	Part	Sheetname	Sheet#	Filename
55	R19	100	YRCVSELECT	3	YRCVSEL.SCH
56	R20	100	YRCVSELECT	3	YRCVSEL.SCH
57	R21	100	YRCVSELECT	3	YRCVSEL.SCH
58	R22	3.3K	RS232	5	RS232.SCH
59	R23	3.3K	RS232	5	RS232.SCH
60	R24	1K	RS232	5	RS232.SCH
61	R25	1.8K	RS232	5	RS232.SCH
62	R26	22K	RS232	5	RS232.SCH
63	R27	22K	RS232	5	RS232.SCH
64	R28	1K	RS232	5	RS232.SCH
65	R29	1.8K	RS232	5	RS232.SCH
66	R30	22K	RS232	5	RS232.SCH
67	R31	22K	RS232	5	RS232.SCH
68	R32	10K	YDRIVER	4	YDRIVER.SCH
69	R33	10K	YDRIVER	4	YDRIVER.SCH
70	R34	10K	YDRIVER	4	YDRIVER.SCH
71	R35	10K	YDRIVER	4	YDRIVER.SCH
72	R36	10K	YDRIVER	4	YDRIVER.SCH
73	R37	10K	YDRIVER	4	YDRIVER.SCH
74	R38	10K	YDRIVER	4	YDRIVER.SCH
75	R39	10K	YDRIVER	4	YDRIVER.SCH
76	R40	2.2	YDRIVER	4	YDRIVER.SCH
77	R41	4.7K	YDRIVER	4	YDRIVER.SCH
78	R42	4.7K	YDRIVER	4	YDRIVER.SCH
79	R43	470K	YDRIVER	4	YDRIVER.SCH
80	R44	100	YDRIVER	4	YDRIVER.SCH
81	R45	100	YDRIVER	4	YDRIVER.SCH
82	R46	100	YDRIVER	4	YDRIVER.SCH
83	R47	100	YDRIVER	4	YDRIVER.SCH
84	R48	100	YDRIVER	4	YDRIVER.SCH
85	R49	100	YDRIVER	4	YDRIVER.SCH
86	R50	100	YDRIVER	4	YDRIVER.SCH
87	R51	100	YDRIVER	4	YDRIVER.SCH
88	R52	15K	YDRIVER	4	YDRIVER.SCH
89	R53	15K	YDRIVER	4	YDRIVER.SCH
90	R54	1M	YDRIVER	4	YDRIVER.SCH
91	R55	1M	YDRIVER	4	YDRIVER.SCH
92	R56	18K	YDRIVER	4	YDRIVER.SCH
93	R57	22K	YDRIVER	4	YDRIVER.SCH
94	R58	22K	RS232	5	RS232.SCH
95	R59	22K	RS232	5	RS232.SCH
96	SW1	SW PUSHBUTTON	<<<root>>>	1	CONTROL.SCH
97	SW2	SW DIP-8	ID-UNIT	2	IDUNIT.SCH
98	U1	14099	<<<root>>>	1	CONTROL.SCH
99	U2	14099	<<<root>>>	1	CONTROL.SCH
100	U3	14099	<<<root>>>	1	CONTROL.SCH
101	U4	14515	<<<root>>>	1	CONTROL.SCH
102	U5	MC14049UB	<<<root>>>	1	CONTROL.SCH

103	U6	MC14049UB	<<<root>>>	1	CONTROL.SCH
104	U7	MC14049UB	<<<root>>>	1	CONTROL.SCH
105	U8	MC14069UB	<<<root>>>	1	CONTROL.SCH
106	U9	68HC05C4	<<<root>>>	1	CONTROL.SCH
107	U10A	14069	<<<root>>>	1	CONTROL.SCH
108	U10B	14069	<<<root>>>	1	CONTROL.SCH

SCANNER CONTROLER — Revised: April 19, 1992
RIT - CONTROL.SCH — Revision: 1.0
Part Cross Reference Listing — April 19, 1992 20:04:44 — Page 3

Item	Reference	Part	Sheetname	Sheet#	Filename
109	U10C	14069	<<<root>>>	1	CONTROL.SCH
110	U10D	14069	<<<root>>>	1	CONTROL.SCH
111	U10E	14069	RS232	5	RS232.SCH
112	U10F	14069	RS232	5	RS232.SCH
113	U11	4512	ID-UNIT	2	IDUNIT.SCH
114	U12A	CMP-04	RS232	5	RS232.SCH
115	U12B	CMP-04	RS232	5	RS232.SCH
116	U12C	CMP-04	RS232	5	RS232.SCH
117	U12D	CMP-04	RS232	5	RS232.SCH
118	U13A	CMP-04	YDRIVER	4	YDRIVER.SCH
119	U13B	CMP-04	YDRIVER	4	YDRIVER.SCH
120	X1	CRYSTAL 4.0MHz	<<root>>>	1	CONTROL.SCH

TRANSMITTER CARD -
NET LIST
FILE : TRANSMIT.NET
P.1 TO P.2

```
0001 N00001     U1-3  U2-3
0002 N00002     Q1-EMITTER  R1-2
0003 N00003     R9-1  U3-2
0004 N00004     R9-2  Q1-BASE
0005 N00005     Q2-EMITTER  R2-2
0006 N00006     R10-1  U3-4
0007 N00007     R10-2  Q2-BASE
0008 N00008     Q3-EMITTER  R3-2
0009 Q0/1       U3-3  U2-9
0010 Q1/1       U3-5  U2-10
0011 Q2/1       U3-7  U2-11
0012 N00012     U3-6  R11-1
0013 N00013     R11-2  Q3-BASE
0014 Q3/1       U3-9  U2-12
0015 N00015     U3-10  R12-1
0016 N00016     Q4-EMITTER  R4-2
0017 Q4/1       U3-11  U2-13
0018 N00018     U3-12  R13-1
0019 Q5/1       U3-14  U2-14
0020 N00020     U3-15  R14-1
0021 N00021     R12-2  Q4-BASE
0022 N00022     Q5-EMITTER  R5-2
0023 N00023     R13-2  Q5-BASE
0024 N00024     Q6-EMITTER  R6-2
0025 N00025     R14-2  Q6-BASE
0026 Q6/1       U4-3  U2-15
0027 N00027     U4-2  R15-1
0028 N00028     Q7-EMITTER  R7-2
0029 Q7/1       U4-5  U2-1
0030 N00030     U4-4  R16-1
0031 N00031     R15-2  Q7-BASE
0032 N00032     R8-2  Q8-EMITTER
0033 N00033     R16-2  Q8-BASE
0034 TID3       R17-2  U1-14
0035 TID2       R18-2  U1-1
0036 TA3        U1-10
0037 +VH
0038 TID1       R20-2  U1-9
0039 TA4        U1-7
0040 TID0       R19-2  U1-11
0041 TA5        U1-2
0042 TA6        U1-15
0043 TA0        U2-5
0044 TA1        U2-6
0045 TA2        U2-7
0046 TEN        U1-6
0047 -VH        RPACK1-16  RPACK1-15  RPACK1-14  RPACK1-13
                RPACK1-12  RPACK1-11  RPACK1-10  RPACK1-9
0048 GND        D1-ANODE  D2-ANODE  D3-ANODE  D4-ANODE
                D5-ANODE  D6-ANODE  D7-ANODE  D8-ANODE
                U4-8  U4-7  U4-9  U4-11
                U4-14  U3-8  U2-8  U1-8
                U1-5  U2-4  U1-4
0049 VCC        R8-1  R7-1  U4-1  R6-1
                R5-1  R4-1  R3-1  U3-1
```

1

```
                                        R2-1  R1-1  U2-2  U2-16
                                        R17-1  R18-1  R20-1  R19-1
                                        U1-16
     0050 XDRV0                         D1-CATHODE  RPACK1-1  Q1-COLLECTOR
     0051 XDRV1                         D2-CATHODE  Q2-COLLECTOR  RPACK1-2
     0052 XDRV2                         D3-CATHODE  Q3-COLLECTOR  RPACK1-3
     0053 XDRV3                         D4-CATHODE  Q4-COLLECTOR  RPACK1-4
     0054 XDRV4                         D5-CATHODE  Q5-COLLECTOR  RPACK1-5
     0055 XDRV5                         D6-CATHODE  Q6-COLLECTOR  RPACK1-6
     0056 XDRV6                         D7-CATHODE  Q7-COLLECTOR  RPACK1-7
     0057 XDRV7                         D8-CATHODE  Q8-COLLECTOR  RPACK1-8
     0058 AYDRV0  OYDRV0
     0059 AYDRV1  OYDRV1
     0060 AYDRV2  OYDRV2
     0061 AYDRV3  OYDRV3
     0062 AYDRV4  OYDRV4
     0063 AYDRV5  OYDRV5
     0064 AYDRV6  OYDRV6
     0065 AYDRV7  OYDRV7
```

TRANSMITTER CARD
CROSS REF.
FILE: TRANSMIT.CRF
R.1 TO P2

TRANSMITTER Revised: April 19, 1992
RIT - TRANSMIT.SCH Revision: 1.0
Part Cross Reference Listing April 19, 1992 18:40:52 Page 1

Item	Reference	Part	Sheetname	Sheet#	Filename
1	D1	1N4150	<<<root>>>	1	TRANSMIT.SCH
2	D2	1N4150	<<<root>>>	1	TRANSMIT.SCH
3	D3	1N4150	<<<root>>>	1	TRANSMIT.SCH
4	D4	1N4150	<<<root>>>	1	TRANSMIT.SCH
5	D5	1N4150	<<<root>>>	1	TRANSMIT.SCH
6	D6	1N4150	<<<root>>>	1	TRANSMIT.SCH
7	D7	1N4150	<<<root>>>	1	TRANSMIT.SCH
8	D8	1N4150	<<<root>>>	1	TRANSMIT.SCH
9	Q1	2N4403	<<<root>>>	1	TRANSMIT.SCH
10	Q2	2N4403	<<<root>>>	1	TRANSMIT.SCH
11	Q3	2N4403	<<<root>>>	1	TRANSMIT.SCH
12	Q4	2N4403	<<<root>>>	1	TRANSMIT.SCH
13	Q5	2N4403	<<<root>>>	1	TRANSMIT.SCH
14	Q6	2N4403	<<<root>>>	1	TRANSMIT.SCH
15	Q7	2N4403	<<<root>>>	1	TRANSMIT.SCH
16	Q8	2N4403	<<<root>>>	1	TRANSMIT.SCH
17	RPACK1	10K	<<<root>>>	1	TRANSMIT.SCH
18	R1	33	<<<root>>>	1	TRANSMIT.SCH
19	R2	33	<<<root>>>	1	TRANSMIT.SCH
20	R3	33	<<<root>>>	1	TRANSMIT.SCH
21	R4	33	<<<root>>>	1	TRANSMIT.SCH
22	R5	33	<<<root>>>	1	TRANSMIT.SCH
23	R6	33	<<<root>>>	1	TRANSMIT.SCH
24	R7	33	<<<root>>>	1	TRANSMIT.SCH
25	R8	33	<<<root>>>	1	TRANSMIT.SCH
26	R9	100	<<<root>>>	1	TRANSMIT.SCH
27	R10	100	<<<root>>>	1	TRANSMIT.SCH
28	R11	100	<<<root>>>	1	TRANSMIT.SCH
29	R12	100	<<<root>>>	1	TRANSMIT.SCH
30	R13	100	<<<root>>>	1	TRANSMIT.SCH
31	R14	100	<<<root>>>	1	TRANSMIT.SCH
32	R15	100	<<<root>>>	1	TRANSMIT.SCH
33	R16	100	<<<root>>>	1	TRANSMIT.SCH
34	R17	22K	<<<root>>>	1	TRANSMIT.SCH
35	R18	22K	<<<root>>>	1	TRANSMIT.SCH
36	R19	22K	<<<root>>>	1	TRANSMIT.SCH
37	R20	22K	<<<root>>>	1	TRANSMIT.SCH
38	U1	14585	<<<root>>>	1	TRANSMIT.SCH
39	U2	4099	<<<root>>>	1	TRANSMIT.SCH
40	U3	MC14049UB	<<<root>>>	1	TRANSMIT.SCH
41	U4	MC14049UB	<<<root>>>	1	TRANSMIT.SCH

TRANSMITTER Revised: April 19, 1992
RIT - TRANSMIT.SCH Revision: 1.0
Part Cross Reference Listing April 19, 1992 18:40:52 Page 2

Item	Part	Reference	Sheetname	Sheet#	Filename
1	100	R9	<<<root>>>	1	TRANSMIT.SCH
2	100	R10	<<<root>>>	1	TRANSMIT.SCH
3	100	R11	<<<root>>>	1	TRANSMIT.SCH
4	100	R12	<<<root>>>	1	TRANSMIT.SCH
5	100	R13	<<<root>>>	1	TRANSMIT.SCH
6	100	R14	<<<root>>>	1	TRANSMIT.SCH
7	100	R15	<<<root>>>	1	TRANSMIT.SCH
8	100	R16	<<<root>>>	1	TRANSMIT.SCH
9	10K	RPACK1	<<<root>>>	1	TRANSMIT.SCH
10	14585	U1	<<<root>>>	1	TRANSMIT.SCH
11	1N4150	D1	<<<root>>>	1	TRANSMIT.SCH
12	1N4150	D2	<<<root>>>	1	TRANSMIT.SCH
13	1N4150	D3	<<<root>>>	1	TRANSMIT.SCH
14	1N4150	D4	<<<root>>>	1	TRANSMIT.SCH
15	1N4150	D5	<<<root>>>	1	TRANSMIT.SCH
16	1N4150	D6	<<<root>>>	1	TRANSMIT.SCH
17	1N4150	D7	<<<root>>>	1	TRANSMIT.SCH
18	1N4150	D8	<<<root>>>	1	TRANSMIT.SCH
19	22K	R17	<<<root>>>	1	TRANSMIT.SCH
20	22K	R18	<<<root>>>	1	TRANSMIT.SCH
21	22K	R19	<<<root>>>	1	TRANSMIT.SCH
22	22K	R20	<<<root>>>	1	TRANSMIT.SCH
23	2N4403	Q1	<<<root>>>	1	TRANSMIT.SCH
24	2N4403	Q2	<<<root>>>	1	TRANSMIT.SCH
25	2N4403	Q3	<<<root>>>	1	TRANSMIT.SCH
26	2N4403	Q4	<<<root>>>	1	TRANSMIT.SCH
27	2N4403	Q5	<<<root>>>	1	TRANSMIT.SCH
28	2N4403	Q6	<<<root>>>	1	TRANSMIT.SCH
29	2N4403	Q7	<<<root>>>	1	TRANSMIT.SCH
30	2N4403	Q8	<<<root>>>	1	TRANSMIT.SCH
31	33	R1	<<<root>>>	1	TRANSMIT.SCH
32	33	R2	<<<root>>>	1	TRANSMIT.SCH
33	33	R3	<<<root>>>	1	TRANSMIT.SCH
34	33	R4	<<<root>>>	1	TRANSMIT.SCH
35	33	R5	<<<root>>>	1	TRANSMIT.SCH
36	33	R6	<<<root>>>	1	TRANSMIT.SCH
37	33	R7	<<<root>>>	1	TRANSMIT.SCH
38	33	R8	<<<root>>>	1	TRANSMIT.SCH
39	4099	U2	<<<root>>>	1	TRANSMIT.SCH
40	MC14049UB	U3	<<<root>>>	1	TRANSMIT.SCH
41	MC14049UB	U4	<<<root>>>	1	TRANSMIT.SCH

RECEIVER CARD
NETLIST
FILE RCV.NET
P.1 TO P.2

```
0001 N00001     U1-3  U2-3
0002 N00002     Q1-EMITTER  R1-2
0003 N00003     R9-1  U3-2
0004 N00004     R9-2  Q1-BASE
0005 N00005     Q2-EMITTER  R2-2
0006 N00006     R10-1  U3-4
0007 N00007     R10-2  Q2-BASE
0008 N00008     Q3-EMITTER  R3-2
0009 Q0/1       U3-3  U2-9
0010 Q1/1       U3-5  U2-10
0011 Q2/1       U3-7  U2-11
0012 N00012     U3-6  R11-1
0013 N00013     R11-2  Q3-BASE
0014 Q3/1       U3-9  U2-12
0015 N00015     U3-10  R12-1
0016 N00016     Q4-EMITTER  R4-2
0017 Q4/1       U3-11  U2-13
0018 N00018     U3-12  R13-1
0019 Q5/1       U3-14  U2-14
0020 N00020     U3-15  R14-1
0021 N00021     R12-2  Q4-BASE
0022 N00022     Q5-EMITTER  R5-2
0023 N00023     R13-2  Q5-BASE
0024 N00024     Q6-EMITTER  R6-2
0025 N00025     R14-2  Q6-BASE
0026 Q6/1       U4-3  U2-15
0027 N00027     U4-2  R15-1
0028 N00028     Q7-EMITTER  R7-2
0029 Q7/1       U4-5  U2-1
0030 N00030     U4-4  R16-1
0031 N00031     R15-2  Q7-BASE
0032 N00032     R8-2  Q8-EMITTER
0033 N00033     R16-2  Q8-BASE
0034 TID3       R17-2  U1-14
0035 TID2       R18-2  U1-1
0036 TA3        U1-10
0037 +VH
0038 TID1       R20-2  U1-9
0039 TA4        U1-7
0040 TID0       R19-2  U1-11
0041 TA5        U1-2
0042 TA6        U1-15
0043 TA0        U2-5
0044 TA1        U2-6
0045 TA2        U2-7
0046 TEN        U1-6
0047 -VH        RPACK1-16  RPACK1-15  RPACK1-14  RPACK1-13
                RPACK1-12  RPACK1-11  RPACK1-10  RPACK1-9
0048 GND        D1-ANODE  D2-ANODE  D3-ANODE  D4-ANODE
                D5-ANODE  D6-ANODE  D7-ANODE  D8-ANODE
                U4-8  U4-7  U4-9  U4-11
                U4-14  U3-8  U2-8  U1-8
                U1-5  U2-4  U1-4
0049 VCC        R8-1  R7-1  U4-1  R6-1
                R5-1  R4-1  R3-1  U3-1
```

1

```
                                    R2-1  R1-1  U2-2  U2-16
                                    R17-1  R18-1  R20-1  R19-1
                                    U1-16
     0050 XDRV0                     D1-CATHODE  RPACK1-1  Q1-COLLECTOR
     0051 XDRV1                     D2-CATHODE  Q2-COLLECTOR  RPACK1-2
     0052 XDRV2                     D3-CATHODE  Q3-COLLECTOR  RPACK1-3
     0053 XDRV3                     D4-CATHODE  Q4-COLLECTOR  RPACK1-4
     0054 XDRV4                     D5-CATHODE  Q5-COLLECTOR  RPACK1-5
     0055 XDRV5                     D6-CATHODE  Q6-COLLECTOR  RPACK1-6
     0056 XDRV6                     D7-CATHODE  Q7-COLLECTOR  RPACK1-7
     0057 XDRV7                     D8-CATHODE  Q8-COLLECTOR  RPACK1-8
     0058 AYDRV0  OYDRV0
     0059 AYDRV1  OYDRV1
     0060 AYDRV2  OYDRV2
     0061 AYDRV3  OYDRV3
     0062 AYDRV4  OYDRV4
     0063 AYDRV5  OYDRV5
     0064 AYDRV6  OYDRV6
     0065 AYDRV7  OYDRV7
```

RECEIVER CARD
CROSS REF
FILE RCV CRF
P.1 TO P.

RECIEVER Revised: April 17, 1992
RIT - RCV.SCH Revision: 1.0
Part Cross Reference Listing April 19, 1992 17:58:37 Page 1

Item	Reference	Part	Sheetname	Sheet#	Filename
1	C1	47uF	<<<root>>>	1	RCV.SCH
2	C2	10nF	<<<root>>>	1	RCV.SCH
3	C3	47uF	<<<root>>>	1	RCV.SCH
4	C4	1000uF	<<<root>>>	1	RCV.SCH
5	C5	10nF	<<<root>>>	1	RCV.SCH
6	C6	0.1uF	RCVAMP	2	RCVAMP.SCH
7	C7	100pF	RCVAMP	2	RCVAMP.SCH
8	C8	0.1uF	RCVAMP	3	RCVAMP.SCH
9	C9	100pF	RCVAMP	3	RCVAMP.SCH
10	C10	0.1uF	RCVAMP	4	RCVAMP.SCH
11	C11	100pF	RCVAMP	4	RCVAMP.SCH
12	C12	0.1uF	RCVAMP	5	RCVAMP.SCH
13	C13	100pF	RCVAMP	5	RCVAMP.SCH
14	D1	1N4150	RCVAMP	2	RCVAMP.SCH
15	D2	1N4150	RCVAMP	3	RCVAMP.SCH
16	D3	1N4150	RCVAMP	4	RCVAMP.SCH
17	D4	1N4150	RCVAMP	5	RCVAMP.SCH
18	R1	330	<<<root>>>	1	RCV.SCH
19	R2	18K	<<<root>>>	1	RCV.SCH
20	R3	220	<<<root>>>	1	RCV.SCH
21	R4	220	<<<root>>>	1	RCV.SCH
22	R5	220	<<<root>>>	1	RCV.SCH
23	R6	220	<<<root>>>	1	RCV.SCH
24	R7	220	<<<root>>>	1	RCV.SCH
25	R8	750K	<<<root>>>	1	RCV.SCH
26	R9	27K	RCVAMP	2	RCVAMP.SCH
27	R10	330	RCVAMP	2	RCVAMP.SCH
28	R11	330	RCVAMP	2	RCVAMP.SCH
29	R12	10K	RCVAMP	2	RCVAMP.SCH
30	R13	3.9K	RCVAMP	2	RCVAMP.SCH
31	R14	560	RCVAMP	2	RCVAMP.SCH
32	R15	3.9K	RCVAMP	2	RCVAMP.SCH
33	R16	1.1M	RCVAMP	2	RCVAMP.SCH
34	R17	27K	RCVAMP	3	RCVAMP.SCH
35	R18	330	RCVAMP	3	RCVAMP.SCH
36	R19	330	RCVAMP	3	RCVAMP.SCH
37	R20	10K	RCVAMP	3	RCVAMP.SCH
38	R21	3.9K	RCVAMP	3	RCVAMP.SCH
39	R22	560	RCVAMP	3	RCVAMP.SCH
40	R23	3.9K	RCVAMP	3	RCVAMP.SCH
41	R24	1.1M	RCVAMP	3	RCVAMP.SCH
42	R25	27K	RCVAMP	4	RCVAMP.SCH
43	R26	330	RCVAMP	4	RCVAMP.SCH
44	R27	330'	RCVAMP	4	RCVAMP.SCH
45	R28	10K	RCVAMP	4	RCVAMP.SCH
46	R29	3.9K	RCVAMP	4	RCVAMP.SCH
47	R30	560	RCVAMP	4	RCVAMP.SCH
48	R31	3.9K	RCVAMP	4	RCVAMP.SCH

1

```
49  R32        1.1M     RCVAMP        4     RCVAMP.SCH
```

RECIEVER Revised: April 17, 1992
RIT - RCV.SCH Revision: 1.0
Part Cross Reference Listing April 19, 1992 17:58:37 Page 2

Item	Reference	Part	Sheetname	Sheet#	Filename
50	R33	27K	RCVAMP	5	RCVAMP.SCH
51	R34	330	RCVAMP	5	RCVAMP.SCH
52	R35	330	RCVAMP	5	RCVAMP.SCH
53	R36	10K	RCVAMP	5	RCVAMP.SCH
54	R37	3.9K	RCVAMP	5	RCVAMP.SCH
55	R38	560	RCVAMP	5	RCVAMP.SCH
56	R39	3.9K	RCVAMP	5	RCVAMP.SCH
57	R40	1.1M	RCVAMP	5	RCVAMP.SCH
58	U1	14503	<<<root>>>	1	RCV.SCH
59	U2A	CMP-04	RCVAMP	2	RCVAMP.SCH
60	U2B	CMP-04	RCVAMP	3	RCVAMP.SCH
61	U2C	CMP-04	RCVAMP	4	RCVAMP.SCH
62	U2D	CMP-04	RCVAMP	5	RCVAMP.SCH
63	U3	OP16	RCVAMP	2	RCVAMP.SCH
64	U4	OP16	RCVAMP	3	RCVAMP.SCH
65	U5	OP16	RCVAMP	4	RCVAMP.SCH
66	U6	OP16	RCVAMP	5	RCVAMP.SCH

ANNEX II

MICROCONTROLLER SOFTWARE

FILE C4.ASM

```
* I/O ports:
PRTA    equ     $0
PRTB    equ     $1
PRTC    equ     $2
PRTD    equ     $3
DDRA    equ     $4      ; Data Direction
DDRB    equ     $5
DDRC    equ     $6

* Serial Communication:
SCBRR   equ     $0d     ; SC Buad Rate
SCCR1   equ     $0e     ; SC Control Register 1
SCCR2   equ     $0f     ; SC Control Register 2
SCSR    equ     $10     ; SC Status Register
SCDR    equ     $11     ; SC Date Register

* Timer:
TCR     equ     $12     ; Timer Control Register
TSR     equ     $13     ; Timer Status Register
ICHR    equ     $14     ; Input Capture High Register
ICLR    equ     $15     ; Input Capture Low  Register
TOCMPH  equ     $16     ; Output Capture High Register
TOCMPL  equ     $17     ; Output Capture Low  Register
ICH     equ     $18     ; Counter High Reg.
TCL     equ     $19     ; Counter Low  Reg.
TACH    equ     $1A     ; Alternate Counter High Reg.
TACL    equ     $1B     ; Alternate Counter Low  Reg.

* reset and interupts vectors:
VRESET  equ     $1ffe   ; Reset
VSWI    equ     $1ffc   ; Softwart Interupt (SWI)
VIRQ    equ     $1ffa   ; External Interupt
VTIMER  equ     $1ff8   ; Timer Interupts (overflow/input/output)
VSCI    equ     $1ff6   ; Serital communication interupt
```

MICROCONTROLLER SOFTWARE FILE SCAN.ASM P 1 - P 34

```
$set DEBUG
$base 10T

$include "c4.asm"        ;definition of labels

; Time defenitions:
RXREL       equ 2500     ; Relaxing time after selecting reciever row
                         ; RXSEL*2uSEC (16 to 65535)
TXREL       equ 80       ; Relaxing time after transmitting

; Memory defenitions:
INSTUSESIZE equ 63       ; Ram for various commands use
SCBUFSIZE   equ 56       ; Size of SC data transmit buffer
INSTBUFSIZE equ 12       ; Size of SC instruction Recieve buffer
NDSR        equ 4        ; bit 4 in PORT C indicates Not Data Set Ready
NDTR        equ 5        ; bit 5 in PORT C indicates Not Data Terminal Ready

*************************************
;RAM ADDRESSES - VARIOUS VARIABLES
*************************************
        ORG $50          ; (OVERLAP BOTTOM OF STACK SPACE)
ID      ds 1             ; MCU ID NUMBER
RXSR    ds 1             ; RX cards Status Register
                         ; Bit 0 Set if EXTENDED addressing MODE needed
                         ;       (cards beyond 0L,0H,1L,1H present)
                         ; Bit 1 Set if Cards not present by order
                         ;       (For example card 3L connected ,but not 2H)
                         ; Bit 6 Set if found any card with one or more of the
                         ;       lines not connected
* SCI
SCITA   ds 1             ; points the next SC transmit address from
                         ; the bottom of transmit buffer
                         ; set call to SENDDATA when DTR
SCITEA ds 1              ; points end address of SC transmit from bottom
                         ; of buffer
SCBUF ds SCBUFSIZE       ; address of SC data transmit buffer
                         ; bit 7 should always be 0
INSTBUF ds INSTBUFSIZE   ; Address of instruction recieved
                         ; byte 0 - bits 0-3 location in inst. buffer for next
                         ;                   data recieve
                         ; byte 1 - bits 0-4 instruction code
                         ;          bit  5   error while recieving instruction
                         ;          bit  6   Ready for next instruction
                         ;          bit  7   set after recieved all instruction
                         ; byte 2 - bits 0-3 location of last byte of message
                         ;                   relative to bottom of inst. buffer
                         ; bytes 3-15 - data for each instruction
INSTUSE ds INSTUSESIZE   ; This byte are used for various command by MCU
                         ; each command use it in different way
* GENERAL
TSTORE ds 1              ; bit 0 - store I flag by STOREI subroutine
                         ; bits 1-7 UNUSED
QTMP1   ds 1             ; byte for temporary data in interrupts
TMP1    ds 1             ; 2 bytes for passing data to subroutines
TMP2    ds 1             ; or teporary data
```

```
* COMMANDS USE IN INSTUSE RAM
; Recieved Instruction code location
CMDCODE     equ INSTBUF+1

; GETSTAT (8) and GETWIDTH (12)
WAVE        equ INSTUSE       ; to INSTUSE+29, Wave in reciever
WBITNUM     equ INSTUSE+30
STATCOUNT   equ INSTUSE+40


; SCANCONFIG (10)
; The following used by GETCONNECT subroutine:
SAMPLEA     equ INSTUSE
SAMPLEB     equ SAMPLEA+1
SAMPLEC     equ SAMPLEB+1
RCVPRESENT  equ SAMPLEC+1
PPSTAT      equ RCVPRESENT+1
PPCONNECT   equ PPSTAT+1
PPWARN      equ PPCONNECT+1
; The following used by SCANCONFIG and SCANHALFTX:
HALFTA      equ PPWARN+1
TOHALFTA    equ HALFTA+1      ;(only by SCANCONFIG)
TXCOL       equ TOHALFTA+1
RXCOL       equ TXCOL+1
RXROW       equ RXCOL+1
RENADDR     equ RXROW+1
HTXSTATUS   equ RENADDR+1
DATALADDR   equ HTXSTATUS+1
HTXDATAL    equ DATALADDR+1    ;32 bytes
HTXDATAH    equ HTXDATAL+32    ;16 bytes

;
;The following used by REPRXCARDS (and REP2RXCARD) subroutines:
REPRXCOL    equ INSTUSE
REPRXADDR   equ REPRXCOL+1
TOREPRXADDR  equ REPRXADDR+1
BYRCV       equ TOREPRXADDR+1
PYRCV       equ BYRCV+1      ;8 bytes

************************************
;MACROS
************************************
$MACRO GO_TIMER TIME              ; START TIMER TO TIME VALUE
               LDA #(%1>8)        ; TIME HI
               LDX #(%1&$FF)      ; TIME LO
               JSR NTTIMER        ; Set oscilation relax time
$MACROEND
$MACRO WAIT_TIMER                 ; Wait until timer time enpalsed
               BRCLR 6,TSR,*      ; Wait for timer output compare
$MACROEND

************************************
;VECTORS DEFINITION
```

```
**********************************
            org VRESET
            dw  QRESET
            org VTIMER   ;(Not in use)
            dw  QTIMER
            org VSCI
            dw  QSCI


            ORG $100
*****************************************************
;RESET SIGNAL
*****************************************************
QRESET:     RSP
            SEI
            LDA #$00     ; Set data directions of all ports
            STA DDRA
            LDA #$ff
            STA DDRB
            LDA #$5f
            STA DDRC
            BSET 5,PRTC  ; SC Not ready to recieve data - yet
            JSR LAINIT   ; Latch initialization
            CLRA
            LDX #3
            JSR LATCHWR  ; I/O CARD latch=$00 (latch 3)
            BSET 2,PRTC  ; SCAN signal off
            JSR GETID    ; GET ID NUMBER OF MCU
            JSR SCINIT   ; INIT Serial Communication
            JMP START

START:      CLI
            JMP ROUTINE

*****************************************************
* This is the main program which wait for instruction
* or other change when controller idle
*****************************************************
ROUTINE     CLI
            JSR LAINIT       ; Reset latches to prevent transistor being on
            JSR READDATA     ; Read instruction from terminal
            LDA CMDCODE
            CMP #$40         ; Check if recieved new command
            BEQ NONEW
JEXECUTE    JSR EXECUTE      ; New command execution
NONEW       SEI
            JSR SENDDATA     ; send only if possible and needed
            CLI
            JMP ROUTINE
EXECUTE     BRSET 5;CMDCODE,ERROR  ; Error while recieving command
            LDA CMDCODE
            CMP #$80          ;Compare stop
            BEQ CSTOP
            CMP #$81          ;Compare reset
```

```
                BEQ CRESET
                CMP #$88          ;Compare getstat
                BEQ CGETSTAT
                CMP #$89          ;Compare testpoint
                BEQ CTESTPOINT
                CMP #$8A          ;Compare to scanconfig
                BEQ CSCANCONFIG
                CMP #$8B          ;Compare to reprxcards
                BEQ CREPRXCARDS
                CMP #$8C          ;Compare to getwidth command
                BEQ CGETWIDTH
                BRA ERROR             ; Command not recognized, go error
ERROR           LDA #$80          ;Start transmit
                JSR SCFAPPEND
                LDA ID            ;transmit ID
                JSR SCFAPPEND
                LDA #$20          ;Transmit error in recieving mark
                JSR SCFAPPEND
                LDA CMDCODE       ;Transmit command code which error occured
                JSR SCFAPPEND
                CLRA              ;End of message
                JSR SCFAPPEND
                CLR INSTBUF
                LDA #$40          ;Reset Instruction buffer
                STA CMDCODE
                RTS

CSTOP           JMP STOP
CRESET          JMP RESET
CGETSTAT        JMP GETSTAT
CTESTPOINT      JMP TESTPOINT
CSCANCONFIG     JMP SCANCONFIG
CREPRXCARDS     JMP REPRXCARDS
CGETWIDTH       JMP GETWIDTH

*************************
; STOP command
; If SCBUF Not empty Remove content of SCBUF and send BREAK to terminal
STOP            SEI
                LDA SCITA
                CMP SCITEA
                BEQ STOPTEMPTY    ;If TX empty skip break send
                BSET SBK,SCCR2    ; Send BREAK
                BCLR SBK,SCCR2
STOPTEMPTY      CLR SCITA
                CLR SCITEA
                CLI
                CLR INSTBUF       ;Reset instruction buffer
                BSET 6,CMDCODE
                RTS

*************************
;Send break to terminal if transmitter buffer not empty
;and reset the MCU
RESET           JSR STOP
```

```
                JMP QRESET

        ************************
        ; Executing get statics command
        ; Format: ID,8,7,YRCV latch,RCVEN,
        ;         TX Y-DRV latch,TA (bit 7 = 0),Number of waves to create (1 to 256)
        ; Return transmit of:
        ;   128,
        ; for each wave: [ID,8,left to count,
        ;                 point of start transmition pulse,point of end trans. pulse,
        ;                 30,30 bytes of wave from end to start (from PRTA)...],
        ; 0 at end of message
        ; Modifies TMP1,TMP2,AC,XR
GETSTAT       LDX #2            ;Choose y rcv select latch
              LDA INSTBUF+3     ;Yrcv latch (select y of reciever)
              JSR LATCHWR       ;Write ACC to latch
              GO_TIMER RXREL ;Start timer to insure enough space before transmit
              WAIT_TIMER        ;Wait until RXREL time enpalsed
              LDA INSTBUF+4     ;Reciever enable select
              ORA #$F0          ;Prevent unwanted enable of latches
              STA PRTB
              BSET 0,PRTC       ;Enable 4->16 multiplexer latch
              LDX #1            ;Choose Transmitter y-driver latch (short delay)
              BCLR 0,PRTC       ;Disable multiplexer latch (after short delay)
              LDA INSTBUF+5     ; Y-drv latch data
              JSR LATCHWR       ; Write it to latch
              LDX #0            ;Choose transmitter latch
              LDA INSTBUF+6     ;Transmitter latch data
              JSR LATCHWR       ;Choose transmitter
              LDX INSTBUF+7
              STX STATCOUNT
              BSET 6,INSTBUF+1 ;Mark that ready to recieve next instruction
              LDA #$80          ; Send 128 at start of message
              JSR SCFAPPEND     ; Send it
GETSTATLOOP   SEI
              JSR GETPOINT      ; Getting wave
              CLI
              GO_TIMER TXREL ; Minimum time before next transmit
              WAIT_TIMER
              LDA ID
              JSR SCFAPPEND     ; Send ID of card
              LDA #8
              JSR SCFAPPEND     ; Send command code
              LDA STATCOUNT
              JSR SCFAPPEND     ; Send number of waves left to create
              LDA #1            ; Point of start transmition in wave
              JSR SCFAPPEND     ; Send it
              LDA #5            ; Point of end transmition in wave
              JSR SCFAPPEND     ; Send it
              LDA #6            ; Number of points in wave to send
              JSR SCFAPPEND     ; send it
              TAX
              DECX
SENDWAVELOOP  LDA CMDCODE       ; Load command in instruction buffer
              CMP #$80          ; Compare with stop
```

```
                BEQ GETSTATRTS ; If stop, go to end - (doesn't complete sending)
                LDA WAVE,x     ; Load wave value
                JSR SCFAPPEND  ; Append it to transmitter buffer
                DECX
                BPL SENDWAVELOOP ; Send all wave from end to beggining
                DEC STATCOUNT
                BNE GETSTATLOOP  ; Go to next wave if needed
GETSTATRTS      LDA #0
                JSR SCFAPPEND  ; Mark the end
                RTS

*************************
; Executing GETWIDTH command to measure pulse width
; in point of recieving while transmitting in other point
; Format: ID,12,8,YRCV latch,RCVEN,
;         TX Y-DRV latch,TA (bit 7 = 0),Bit of RL (PRTA) to check,
;         Number of pulses to create (1 to 255)
; Return transmit of:
;   128,ID,12,number of pulses created, { 1 byte for each pulse (W) } ,
; 0 at end of message
; pulse width = 10uSec + 2.5uS*W
; (mininum:10uSEC (W=0)  Maximum:45uSEC (W=14)
; Modifies TMP1,TMP2,AC,XR
GETWIDTH        LDX #2          ;Choose y rcv select latch
                LDA INSTBUF+3   ;Yrcv latch (select y of reciever)
                JSR LATCHWR     ;Write ACC to latch
                LDA INSTBUF+4   ;Reciever enable select
                ORA #$F0        ;Prevent unwanted enable of latches
                STA PRTB
                BSET 0,PRTC     ;Enable 4->16 multiplexer latch
                LDX #1          ;Choose Transmitter y-driver latch (short delay)
                BCLR 0,PRTC     ;Disable multiplexer latch (after short delay)
                LDA INSTBUF+5   ; Y-drv latch data
                JSR LATCHWR     ; Write it to latch
                GO_TIMER RXREL ;Start timer to insure enough space before transmit
                WAIT_TIMER      ;Wait until RXREL time enpalsed
                LDX #0          ;Choose transmitter address latch
                LDA INSTBUF+6   ;Transmitter latch data
                JSR LATCHWR     ;Choose transmitter
                LDA INSTBUF+7   ;Bit of PRTA to check
                STA WBITNUM
                LDX INSTBUF+8
                STX STATCOUNT
                BSET 6,INSTBUF+1 ;Mark that ready to recieve next instruction
                LDA #$80        ; Send 128 at start of message
                JSR SCFAPPEND   ; Send it
                LDA ID
                JSR SCFAPPEND   ; Send ID of card
                LDA #12
                JSR SCFAPPEND   ; Send command code
                LDA STATCOUNT
                JSR SCFAPPEND   ; Send number of waves left to create
GETWIDTHLOOP    SEI
                LDX WBITNUM     ; Bit number (reciever) to check
                JSR GETRCVTIME  ; Getting RX width
```

```
                CLI
                JSR SCFAPPEND   ; Append pulse width to transmitter buffer
                GO_TIMER TXREL ; Minimum time before next transmit
                WAIT_TIMER
                LDA CMDCODE     ; Load command in instruction buffer
                CMP #$80        ; Compare with stop
                BEQ GETPULSERTS ; If stop, go to end - (doesn't complete sending)
                DEC STATCOUNT
                BNE GETWIDTHLOOP  ; Go to next wave if needed
GETPULSERTS     LDA #0
                JSR SCFAPPEND   ; Mark the end
                RTS

*************************
; Getting wave shape in reciever before and after transmition
; before using this subroutine define transmition and recieving point
; store data from WAVE to WAVE+29
; transmitter on before WAVE+12
; transmitter off before WAVE+24
; modifies PRTB,AC
;GETWAVE        LDA #$E7
;               STA PRTB        ;Set address to TEN output, DATA 0
;               BCLR 5,PRTB     ;Enable transmitter latch
;$CYCLE_ADDER_ON                 ;Start counting cycles for .lst file
;               LDA PRTA        ;Read reciever RL port
;               STA WAVE
;               LDA PRTA
;               STA WAVE+1
;               LDA PRTA
;               STA WAVE+2
;               LDA PRTA
;               STA WAVE+3
;               LDA PRTA
;               STA WAVE+4
;               LDA PRTA
;               STA WAVE+5
;               LDA PRTA
;               STA WAVE+6
;               LDA PRTA
;               STA WAVE+7
;               LDA PRTA
;               STA WAVE+8
;               LDA PRTA
;               STA WAVE+9
;               LDA PRTA
;               STA WAVE+10
;               LDA PRTA
;               STA WAVE+11
;               BSET 4,PRTB     ;Transmiter Enable (set TEN)
;               LDA PRTA
;               STA WAVE+12
;               LDA PRTA
;               STA WAVE+13
;               LDA PRTA
;               STA WAVE+14
```

```
;               LDA PRTA
;               STA WAVE+15
;               LDA PRTA
;               STA WAVE+16
;               LDA PRTA
;               STA WAVE+17
;               LDA PRTA
;               STA WAVE+18
;               LDA PRTA
;               STA WAVE+19
;               LDA PRTA
;               STA WAVE+20
;               LDA PRTA
;               STA WAVE+21
;               LDA PRTA
;               STA WAVE+22
;               LDA PRTA
;               STA WAVE+23
;               BCLR 4,PRTB      ;Transmitter Disable (Clear TEN)
;               LDA PRTA
;               STA WAVE+24
;               LDA PRTA
;               STA WAVE+25
;               LDA PRTA
;               STA WAVE+26
;               LDA PRTA
;               STA WAVE+27
;               LDA PRTA
;               STA WAVE+28
;               LDA PRTA
;               STA WAVE+29
;$CYCLE_ADDER_OFF            ;Stop counting cycles for .lst file
;               BSET 5,PRTB  ;Disable transmit latch
;               RTS

*********************
;Subroutine to execute SCANCONFIG command
;scans transmittion cards and return connectivity of all cards
; Format: idle,ID,10,4,from Half Transmitter Address (HALFTA), to HALFTA
; (From .. to means the range of addresses transmitter cards will be scanned
; e.g. if you want to scan all system use from=$00  to=$7C)
; return: idle,128, and for each half Trasmitter:
; [ID,10,HALFTA,HTXSTATUS,HALFTXDATAL,HALFTXDATAH,1 if other cards scanned or
;                                                  0 if end of message]
; see SCANHALFTX for explanation about HALFTA,HTXSTATUS,HALFTXDATAL,HALFTXDATAH
SCANCONFIG   LDA INSTBUF+3
             AND #$7C
             STA HALFTA      ;Set half transmitter address of start loop
             LDA INSTBUF+4
             AND #$7C
             STA TOHALFTA    ;half trasnmitter adderr to end loop
             BSET 6,INSTBUF+1 ;Mark that ready to recieve next instruction
SCANHALOOP   JSR SCANHALFTX   ;Scan half TX card
             LDA #$80         ;send start of card/message byte
             JSR SCFAPPEND
```

```
                LDA ID              ;send ID of this MCU
                JSR SCFAPPEND
                LDA #10             ;Command code
                JSR SCFAPPEND
                LDA HALFTA
                AND #$7C            ;send HALFTA of transmitters scanned
                JSR SCFAPPEND
                LDA HTXSTATUS
                JSR SCFAPPEND       ;Transmit HTXSTATUS of transmitter scanned
                BRSET 7,HTXSTATUS,SKIPSCANSCT  ;if bit 7 is set it means
                                               ;TX card not found so don't
                                               ;transmit HTXDATA
                CLRX
SCANSCT         LDA HTXDATAL,x  ;loop to SC transmit of all HTXDATAL,HTXDATAH
                JSR SCFAPPEND
                INCX
                CPX #48
                BNE SCANSCT
SKIPSCANSCT     LDA HALFTA
                AND #$7C
                CMP TOHALFTA
                BEQ ENDSCAN     ;If scanned last transmitter goto ENDSCAN
                JSR READDATA    ;Read data to check for STOP command
                LDA CMDCODE
                CMP #$80        ;If recieved stop command
                BEQ ENDSCAN     ;Stop scanning
                LDA #1   ; transmit 1 to declare there are more cards to transmit
                         ; it will be transmitted when finish its scanning
                JSR SCFAPPEND
                LDA HALFTA
                ADD #4           ; Next half TX card
                STA HALFTA
                BRA SCANHALOOP   ; Next scan
ENDSCAN         LDA #0           ; Code of no more data to transmit
                JSR SCFAPPEND
                RTS

*************************
; Scanning half transmitter card
; parameter passed to subroutine:
;      HALFTA (address) = address of half transmitter card to scan
;              (bit 0,1 ignored (and may be modified during scanning),
;               bit 2 low/high part of card,
;               bits 3-6 ID of card,
;               bit 7 must be 0)
; Return:
; HTXSTATUS: bits 0-3 noise counter (0= no noise, 15=maximum noise)
;            bit 6 set if and only if found error in reciever card connectivity
;                  (not all RL# Lines not connected)
;            bit 7 is one if and only if no points of transmitter card
;                  connected or no reciever card present (in this case
;                  HTXDATAL not updated)
; HTXDATAL: a byte of data for each transmitter point (32 bytes total)
;           which have the following value:
;           if bit 7 is one found receiving point for the transmitter
```

```
;            and other bits describe the reciever point found:
;              bit 0-1 row (line) number in reciever card
;              bit 2 which reciever card of the two (0=L, 1=H)
;              bit 3-5 REN address
;              bit 6 is set if signal recieved in more than 1 point,
;                    (indicate problem only if bit 3 of HTXDATAH also set)
;              bit 7 is 1
;            the column of recieving point is set in HTXDATAH (see below)
;            if bit 7 is 0 it indicates:
;              data $00 = no recieving point found for this transmitter
;              data $20 = transmittion point not connected
;              data $10 = TOVF (Transmitter current overflow)
; HTXDATAH: half byte of data for each transmittion points (total 16 bytes)
;            bit 0-2 points column number of RX point if connected
;            bit 3 is set if recieved signal in more than one point,
;                  with only small gap in reciever pulse width
;                  (so not sure if point is correct)
; The order of bytes in HTXDATAL is:
; OFFSET    COLUMN OF       ROW OF TRANSMITTION
;          TRANSMITTION     IN HALF CARD
; $00        0                0
; $01        0                1
; $02        0                2
; $03        0                3
; $04        1                0
; $05        1                1
; .
; .
; $1E        7                2
; $1F        7                3
;In HTXDATAH:
; OFFSET   BITS     COLUMN OF       ROW OF TRANSMITTION
;                   TRANSMITTION    IN HALF CARD
; $00      0-3      0               0
; $01      0-3      0               1
; .
; .
; $0F      0-3      3               3
; $00      4-7      4               0
; $01      4-7      4               1
; .
; .
; $0F      4-7      7               3
;
; Other Variables (address) used:
; RXCOL   - Column of recieving (0-7)
; RENADDR - Reciever Enable address (0-7)
; TXCOL   - Column of transmittion (0-7)
; RXROW   - Row of recieveing in side couple RX cards (0-7)
; Modifies: TMP1,TMP2,AC,XR
SCANHALFTX   LDA #$80
             STA HTXSTATUS  ;Bit 7 cleared when first transmit point found
             LDX #48
SCANRSTLP    DECX           ;Loop for reseting HTXDATA
             BMI SCANLPEXIT1
```

```
                CLR HTXDATAL,X
                BRA SCANRSTLP
SCANLPEXIT1     CLR RXCOL       ;Start RX COLUMN LOOP with column 0
RXCOLLOOP       LDX RXCOL       ;Load reciever y-selected
                JSR TWOPOWERX   ;AC=2^XR (For Y-rcv select latch)
                LDX #2          ;Reciever column latch write
                JSR LATCHWR     ;Write AC to latch
                GO_TIMER RXREL ;Start counting RXREL time
                                ;to ensure relaxing of rx point after
                                ;changing selected row and before transmittion
                WAIT_TIMER      ;Wait for relaxing
                CLR RENADDR     ;Start RX ENABLE ADDRESS LOOP with cards 0L,0H
RENADDRLOOP     LDA RENADDR     ;Load RX enable address
                ORA #$F0
                STA PRTB        ;Address of RX to PB[0..3]
                BSET 0,PRTC     ;Write to RX 4->16 multiplexer latch
                NOP             ;Delay to ensure writing
                BCLR 0,PRTC     ;Disable multiplexer latch
                LDA PRTA        ;if cards not connected PRTA=$FF
                COMA            ;Each bit of AC=NOT AC (1's complement)
                BNE SCANHXOK0   ;if connected continue checking
                JMP NEXTRENADDR ;If not connected go to next card
SCANHXOK0       CLR TXCOL       ;Start TX COLUMN LOOP with Column 0
TXCOLLOOP       LDX TXCOL       ;Load transmitter y select
                JSR TWOPOWERX   ;AC=2^XREG (for Y-driver latch)
                LDX #1          ;Ydriver latch write
                JSR LATCHWR     ;Write AC to Ydriver latch
                BCLR 0,HALFTA
                BCLR 1,HALFTA   ;Start row loop of TA with row X00
TXROWLOOP       LDA HALFTA      ; Data to write
                LDX #0          ; to transmitter latch
                JSR LATCHWR     ; Write it to latch
                JSR GETCONNECT  ; Check if transmittion point connected
                                ; to one of 8 selected reciever points
                                ; (The delay between 2 following calls to this
                                ;  subroutine is enough for relaxing of previous
                                ;  transmittion, but beware if you shorting
                                ;  this delay by deleting some commands)
                TST PPWARN      ; Check if has warning (marginaly pulse)
                                ;     on any reciever
                BEQ TXNOWARN    ; if no skip re-checking
                JSR GETCONNECT  ; If it has doubts (WARN) check connection again
TXNOWARN        LDA TXCOL
                LSLA
                LSLA
                STA DATALADDR
                LDA HALFTA
                AND #$03
                ORA DATALADDR
                STA DATALADDR ; DATALADDR=COLUMN*4+LINE (offset of HTXDATAL)
                BRCLR 5;PPSTAT,SCANHXOK1    ; Check if transmitter not found
                LDA #$20
                LDX DATALADDR
                STA HTXDATAL,x  ; Transmitter not found bit set
                JMP NEXTTXROW   ; Next row of transmitter
```

```
SCANHXOK1    BCLR 7,HTXSTATUS  ;Clear transmitter card not present bit
                               ;(because found transmittion point connected)
             BRCLR 2,PPSTAT,SCANHXOK1A ; If any reciever card of the couple
                               ; present continue
                               ; (usualy detect cards not present
                               ;    before this command, expect by noise)
             JMP NEXTRENADDR   ;Else goto next card
SCANHXOK1A   BRCLR 3,PPSTAT,SCANHXOK2  ; ERROR in RX cards configuration?
             BSET 6,HTXSTATUS  ;Set ERROR bit
SCANHXOK2    BRCLR 0,PPSTAT,SCANHXOK3   ; Noise detected? no - skip noise increasemen

             LDA HTXSTATUS
             AND #$0F
             CMP #$0F
             BEQ SCANHXOK3  ;Noise counter is maximum
             INC HTXSTATUS  ;Increament noise counter
SCANHXOK3    BRCLR 4,PPSTAT,SCANHXOK4  ; Transmitter current Overflow? (TOVF?)
             LDA #$10
             LDX DATALADDR
             STA HTXDATAL,x ;Transmitter overflow bit set
             BSET 5,HTXSTATUS ;set TOVF in HTXSTATUS
             JMP NEXTTXROW  ;Don't scan this point
SCANHXOK4    CLR RXROW      ;Start check connection from PRTA bit 0
SCANHXLP2    LSR PPCONNECT  ;LOOP Check connection of RX bit by bit
             BCS SCANOKD0
JSHXCONTLP2  JMP SHXCONTLP2 ;If not connected to RX continue
JCHOOSENEW   JMP CHOOSENEW  ;(Used because branch out of range)
SCANOKD0     LDX #40        ;Short delay to ensure relax of previous transmittion
SCANDEL1     DECX
             BNE SCANDEL1
             JSR GETPRTA    ;Wait for PRTA silence
             LDX RXROW      ;Choose bit number for GETRCVTIME
             JSR GETRCVTIME ;Get pulse width in reciever after transmittion
             STA SAMPLEA    ;Store pulse width
             CMP #3         ;Compare pulse width to 17.5uS
             BPL SCANOKD2   ;If 17.5uS or more recieved twice.
                            ;means points connected
             LDA HTXSTATUS  ;Check noise counter
             AND #$0F
             CMP #$0F
             BEQ SCANHXOKD1 ;Noise counter is maximum
             INC HTXSTATUS  ;Increament noise counter
SCANHXOKD1   LDX #70
SCANDEL2     DECX           ;Delay before next transmittion
             BNE SCANDEL2
             JSR GETPRTA    ;Wait PRTA silence
             LDX RXROW      ;Choose bit number for GETRCVTIME
             JSR GETRCVTIME ;Get pulse width in RX point after transmittion
             STA SAMPLEA    ;Store pulse width
             CMP #3         ;Compare it to 17.5uS
             BMI JSHXCONTLP2 ;If shorted than 17.5uS, Ignore pulse
                            ;assuming it is noise
SCANOKD2     LDX DATALADDR  ;Reach SCANOKD2 if point recieved again after
                            ;verifing
             TST HTXDATAL,x
             BEQ JCHOOSENEW ;If No other point already founded
```

```
                                ; goto add put new point in HTXDATA
;Two points founded:
;Get twice pulse width of new point (SAMPLEA = pulse1+pulse2)
;Than twice pulse width of previous point (SAMPLEB = pulse1+pulse2)
;If (SAMPLEA+7.5uS<SAMPLEB) then choose previous point
;If (SAMPLEB+7.5uS<SAMPLEA) then choose new point of RX
;else can't decide, set 2 (or more) points connected flag
                LDX DATALADDR
                LDA HTXDATAL,x
                ORA #$40
                STA HTXDATAL,x ;recieved in more than 1 point bit set
                LDX #100
SCANDEL3        DECX            ;Delay before next transmittion
                BNE SCANDEL3
                JSR GETPRTA     ;Wait PRTA silence
                LDX RXROW       ;Bit number to check in GETRCVTIME
                JSR GETRCVTIME ;AC=pulse width in RX after transmittion
                ADD SAMPLEA     ;Add it to previous pulse width
                STA SAMPLEA     ;SAMPLEA=sum of pulse widths in the 2 samples
                                ;of new connected point
                LDA DATALADDR
                AND #$0F
                TAX
                LDA HTXDATAH,x ;Get column of reciever of previous point
                BRCLR 4,DATALADDR,SCANHXOKD3
                LSRA            ;Rotate left half byte to right
                LSRA
                LSRA
                LSRA
SCANHXOKD3      AND #$07        ;3 bits of column
                TAX
                JSR TWOPOWERX   ;AC=2^XR (set bit of rcv column)
                LDX #2          ;Yrcv latch
                JSR LATCHWR     ;Write to latch
                GO_TIMER RXREL ;Wait point to relax after changing y-rcv
                WAIT_TIMER      ;wait
                LDX DATALADDR
                LDA HTXDATAL,x
                LSRA
                LSRA
                LSRA
                AND #$07        ;AC = Reciever Enable address (RENADDR) of
                                ;previous point
                ORA #$F0
                STA PRTB
                BSET 0,PRTC     ;Enable write REN latch
                NOP
                BCLR 0,PRTC     ;Disable write REN latch
                JSR GETPRTA     ; Wait PRTA silence
                LDX DATALADDR
                LDA HTXDATAL,x ;Get row of recieving in cards (bit of PRTA)
                AND #$07
                TAX
                JSR GETRCVTIME ;Get pulse width in this point
                STA SAMPLEB     ;store pulse width
```

```
                LDX #70
SCANDEL4        DECX            ; Delay before next transmittion
                BNE SCANDEL4
                JSR GETPRTA     ; Wait PRTA silence
                LDX DATALADDR
                LDA HTXDATAL,x ;Get row of recieving in cards (bit of PRTA)
                AND #$07
                TAX
                JSR GETRCVTIME ;Get pulse width in this point (again)
                ADD SAMPLEB     ;add it to previous pulse width in this point
                STA SAMPLEB     ; SAMPLEB = Sum of pulse width in two checks
                LDX RXCOL
                JSR TWOPOWERX
                LDX #2          ;Yrcv latch
                JSR LATCHWR     ;Restore previos yrcv latch
                GO_TIMER RXREL ;Wait point to relax after changing y-rcv
                WAIT_TIMER      ;wait
                LDA RENADDR
                ORA #$F0
                STA PRTB        ;Restore previous latch enable address
                BSET 0,PRTC     ;REN Latch write enable
                NOP
                BCLR 0,PRTC     ;Write disable
; Clear bit of: 2 points connected, undecide which of them
; but set it later if still can't decide which pulse is longer
                LDA DATALADDR
                AND #$0F
                TAX               ;Offset of HTXDATAH
                LDA #$F7
                BRCLR 4,DATALADDR,SCANHXOKD4  ;If right half byte bit 3
                LDA #$7F        ;if Left half byte bit 7
SCANHXOKD4      AND HTXDATAH,x ;Clear bit 3/7
                STA HTXDATAH,x ;Write to correct half byte of TX point
                               ;(without effecting second half byte)
                LDA SAMPLEA
                ADD #3
                CMP SAMPLEB     ; SAMPLEA+3<SAMPLEB ?
                BMI CHOOSEPREV ; If yes, choose previous point of RX
                LDA SAMPLEB
                ADD #3
                CMP SAMPLEA     ; SAMPLEB+3<SAMPLEA ?
                BMI CHOOSENEW  ; If yes, choose new point of RX
; set bit of 2 points connected, undecide which
                LDA DATALADDR
                AND #$0F
                TAX               ;Offset of HTXDATAH
                LDA #$08
                BRCLR 4,DATALADDR,SCANHXOKD5  ;If right half byte bit 3
                LDA #$80        ;if Left half byte bit 7
SCANHXOKD5      ORA HTXDATAH,x ;set bit 3/7
                STA HTXDATAH,x ;Write to correct half byte of TX point
                               ;(without effecting second half byte)
                LDA SAMPLEB
                CMP SAMPLEA     ;SAMPLEB<=SAMPLEA ?
                BLS CHOOSENEW  ;If yes choose new point
```

```
CHOOSEPREV    BRA SHXCONTLP2 ;Leave previous RX point in HTXDATA
CHOOSENEW     LDX DATALADDR  ;offset of HTXDATAL
              LDA RENADDR    ;Put new RX point founded in HTXDATA
              LSLA
              LSLA
              LSLA
              ADD RXROW
              ORA #$80       ;AC=8*RENADDR+RXROW, bit 7 set
              STA TMP1
              LDA HTXDATAL,x
              AND #%01000000 ;Don't change bit 6 (2 points connected)
              ORA TMP1
              STA HTXDATAL,x ;Reciever address and line
              LDA DATALADDR
              AND #$0F
              TAX            ;offset of HTXDATAH
              LDA RXCOL
              BRCLR 4,DATALADDR,SCANHXOK6  ;check which half of byte in HTXDATAH
              LSLA
              LSLA
              LSLA
              LSLA   ;Rotate column to left half byte
SCANHXOK6     STA TMP1        ;Store column (in correct side of byte)
              LDA HTXDATAH,x
              BRCLR 4,DATALADDR,SCANHXOK6A
              AND #$8F        ;Reset 3 bits of column in left half byte
              BRA SCANHXOK6B
SCANHXOK6A    AND #$F8        ;Reset 3 bits of column if right half byte
SCANHXOK6B    ORA TMP1         ;(set correct bits of column in correct side)
              STA HTXDATAH,x   ;Write column received
SHXCONTLP2    LDA PPCONNECT
              BEQ NEXTTXROW    ;If not other point recieved exit loop
              INC RXROW        ;Next row of reciever
              JMP SCANHXLP2
NEXTTXROW     BRCLR 0,HALFTA,SCANHXOK7   ;If TX ROW not reached 4th row than next
              BRCLR 1,HALFTA,SCANHXOK7
              BRA NEXTTXCOL   ;Finished all 4 rows, go to next collumn
SCANHXOK7     INC HALFTA
              JMP TXROWLOOP
NEXTTXCOL     LDX TXCOL
              CPX #7
              BEQ ENDTXCOLLP
              INC TXCOL
              JMP TXCOLLOOP
ENDTXCOLLP    BRSET 7,HTXSTATUS,SCANHTXRTS   ;Scanned all transmittion points
                                             ;But didn't found any connected inductor
                                             ;means TX CARD NOT PRESENT
NEXTRENADDR   LDX RENADDR
              CPX #7
              BEQ NEXTRXCOL  ;No more RX cards for this collumn
                             ;so go to next collumn
              INC RENADDR     ;Next couple of RX cards
              JMP RENADDRLOOP
NEXTRXCOL     LDX RXCOL
              CPX #7
```

```
                BEQ ENDRXCOLLP ;Finished scanning last RX column?
                INC RXCOL      ;No - continue to scan next
                JMP RXCOLLOOP
ENDRXCOLLP      NOP            ;TMP - should be other loop of 2 scans
SCANHTXRTS      RTS

;*************************
; Getting wave at specified point on times:
; before transmittion, 10uS,15.0uS,17.5uS,23.0uS, after stopped transmittion
; before using this subroutine define transmition and recieving point
; store data from WAVE to WAVE+5
; store TF and TOVF (PRTD) to WAVE+6
; transmitter on before WAVE+1
; transmitter off before WAVE+5
; modifies PRTB,AC,XR
GETPOINT        LDA #$E7
                STA PRTB       ;Set address to TEN output, DATA 0
                BCLR 5,PRTB    ;Enable transmitter latch
$CYCLE_ADDER_ON                ;Start counting cycles for .lst file
                LDX #$D7       ;Code of start transmittion
                LDA PRTA       ;Read reciever RL port
                STA WAVE
                STX PRTB       ; 0.0uSec Start transmition
                NOP            ; 1.0uS
                LDX PRTD       ; 2.5uS Check TF and TOVF
                STX WAVE+6     ; 4.5uS
                NOP            ; 5.5uS
                NOP            ; 6.5uS
                NOP            ; 7.5uS
                NOP            ; 8.5uS
                LDA PRTA       ; 10.0uS
                STA WAVE+1     ; 12.0uS
                CLRA           ; 14.5uS
                LDA PRTA       ; 15.0uS
                NOP            ; 16.0uS
                LDX PRTA       ; 17.5uS
                STA WAVE+2     ; 19.5uS
                STX WAVE+3     ; 21.5uS
                LDA PRTA       ; 23.0uS
                STA WAVE+4
                BCLR 4,PRTB    ; Disable trasmitter
                LDA PRTA
                STA WAVE+5     ; 1 Sample after stopping transmitter
$CYCLE_ADDER_OFF               ;Stop counting cycles for .lst file
                BSET 5,PRTB    ;Disable transmit latch
                RTS

;*************************
; Check whether 2 points connected, and if transmitter connected
; before using this subroutine define transmition and recieving point
; IMPORTANT: You should wait between two following calls to this subroutine
;            to ensure relaxing time of transmitter over, and long time (about
;            900 mSEC!) after changing yrcv because of noise come from this
;            transistor switching.
```

```
; This subroutines return the following (by address)
; PPSTAT:  The following bits is set if and only if:
;          bit 0 - a noise in reciever detected before transmittion
;          bit 2 - both Reciever cards not present for this points
;                   (return without doing anything)
;          bit 3 - illegal setting of reciever card, probably
;                  one of the lines to the card disconnected!
;                  (PRTA must be equal to 00 or 0F or F0 or FF
;                   when no transmittion occured)
;          bit 4 - Transmitter driver current overflow (TOVF)
;          bit 5 - Transmitter not found  (not TF)
;                   (return without doing anything)
; each byte of the following has 1 bit for each reciever row,
;     (bit 0 for RL0 ...)
; The signal in reciever at this points:
; SAMPLEA: 10uS, SAMPLEB: 15.0uS, SAMPLEC: 17.5uS
;     (time is measured from start of transmittion)
; PPCONNECT: bit set if reciever and transmitter connected
;                BIT = SAMPLEA*SAMPLEB*SAMPLEC)
; PPWARN:    bit set if not sure if connected, and a second check neccary
;                BIT = SAMPLEA*(SAMPLEB+SAMPLEC)*(NOT PPCONNECT BIT)
; modifies AC,XR,PRTB (=$E7),RCVPRESET,Clear I flag (CLI)
GETCONNECT   CLR PPSTAT      ; Reset values of status,
             CLR PPCONNECT   ; point to point connected
             CLR PPWARN      ; point to point warnning
             JSR GETPRTA     ; Get PRTA before transmittion and wait it
                             ; to be 'silent', also disable interupts!
             TSTX            ; Check if noise detected in PRTA before transmittion
             BEQ GETCONOK1
             BSET 0,PPSTAT   ; If noise detected set noise bit
GETCONOK1    COMA
             STA RCVPRESENT ; Reciever card present if bit equal 1
                             ; check only line which are present
             BNE GETCONOK2   ; at least 1 reciever cards present
RCVNOTPRS    BSET 2,PPSTAT   ; Mark both reciever cards not present
             CLI
             RTS
GETCONOK2    LDA #$E7
             STA PRTB        ;Set address to TEN output, (TEN DATA 0)
             BCLR 5,PRTB     ;Enable transmitter latch
             LDX #$D7        ;Code of start transmittion
             STX PRTB        ; 0.0uSec Start transmition
             NOP             ; 1.0uS
             NOP             ; 2.0uS
             NOP             ; 3.0uS
             NOP             ; 4.0uS
             NOP             ; 5.0uS
             NOP             ; 6.0uS
             NOP             ; 7.0uS
             CLRA            ; 8.5uS    (for Delay only)
             LDA PRTA        ; 10.0uS
             STA SAMPLEA     ; 12.0uS
             CLRA            ; 13.5uS
             LDA PRTA        ; 15.0uS
             NOP             ; 16.0uS
```

```
                LDX PRTA        ; 17.5uS
                STA SAMPLEB
                STX SAMPLEC
                LDA PRTD        ; Check Transmitter Found (TF bit 4)
                AND #$30        ; and Transmitter Overflow (TOVF bit 5)
                EOR #$20        ; reverse TF bit
                ORA PPSTAT      ; add results tp PPSTAT
                STA PPSTAT      ; Set correct bits at PPSTAT
                BEQ GETCONOK3   ; If OK continue, else verify again
                LDA PRTD        ; Check Transmitter Found (TF bit 4)
                AND #$30        ; and Transmitter Overflow (TOVF bit 5)
                EOR #$20        ; reverse TF bit
                ORA PPSTAT      ; add results tp PPSTAT
                STA PPSTAT      ; Set correct bits at PPSTAT
GETCONOK3       BCLR 4,PRTB     ; Disable trasmit point
                BSET 5,PRTB     ; Disable transmit latch
                CLI             ; Enable interrupts again
                LDA SAMPLEB
                ORA SAMPLEC
                AND SAMPLEA
                AND RCVPRESENT  ; (Only for presents RX cards)
                STA PPWARN      ; PPWARN=A*(B+C)  (But later cleared if PPCONNECT)
                LDA SAMPLEA
                AND SAMPLEB
                AND SAMPLEC
                AND RCVPRESENT  ; (Only for presents RX cards)
                STA PPCONNECT   ; PPCONNECT=A*B*C
                COMA
                AND PPWARN
                STA PPWARN      ; PPWARN=PPWARN*(NOT PPCONNECT)
                LDA RCVPRESENT  ; Reciever card present if bit equal 1
                                ; check only line which are present
                CMP #$0F
                BEQ GETCONNRTS  ; only Low reciever card connected
                CMP #$F0
                BEQ GETCONNRTS  ; only High reciever card connected
                CMP #$FF
                BEQ GETCONNRTS  ; Both cards present
                BSET 3,PPSTAT   ; RX hardware failure bit set
GETCONNRTS      RTS

*****************************************************
; Getting value of PRTA by sampling it 4 times
; in delay between the samples, If it's value is not
; changed during 4 samples, it assumes no noise exist
; and return AC=PRTA. if noise detected retry sample it 4 time
; and increase XR (Which counts the noise)
; If after 100 times it can't find constant PRTA it return its value
; Note: Call this subroutine only to get PRTA of time-constant system
;       without any recent transmittion or y-rcv select.
; Return from this subroutine only after 'silence' in PRTA for at least 60uS
; of 1 sample and XR=100
; Modifies: AC=PRTA, XR=noise level, TMP1
```

```
;           Set I-flag (disable interupts)
GETPRTA      LDX #255        ; Noise counter = -1
GETPRTALOOP  CLI             ; Enable interupt for short period
                             ; Any interupt suspended while I-flag set
                             ; will be executed now
             SEI             ; Disable interupts to prevent disturbing
                             ; timing of transmitter-reciever
             LDA PRTA        ; Get PRTA
             INCX            ; increase noise counter
             CPX #100        ; Noise counter reached it's maximum
             BEQ GETPRTARTS  ; Return from loop without getting consist value
                             ; of porta due to hi noise or hardware failure
                             ; XR=254 in this case, AC=PRTA (noisy...)
             CMP PRTA        ; Check if PRTA didn't changed
             BNE GETPRTALOOP  ; If changed re-sample port a
             STX TMP1
             LDX #15
GETPRTADELAY DECX            ; about 45uS delay before next sample
             BNE GETPRTADELAY
             LDX TMP1
             CMP PRTA        ; Check PRTA again
             BNE GETPRTALOOP  ; If changed re-sample it
             NOP
             NOP
             NOP
             NOP
             CMP PRTA         ; Check PRTA again
             BNE GETPRTALOOP  ; If changed re-sample it
GETPRTARTS   RTS              ; PRTA was stable during last 4 samples,
                              ; Return its value in AC assuming no noise occured
                              ; Interupt is disabled now to prevent delay before
                              ; usign of PRTA value
                              : XR is noise counter and equal 0 if no noise detected
*****************************
; Check pulse width in reciever point
; First define the RX and TX latches
; set XR to bit number you want to check (0-7)
; and call this subroutine
; Return pulse width in AC
; pulse time = 10uSEC + AC*2.5uSEC
; if pulse width<=10uSEC AC=0
; if pulse width>=45uSEC AC=14
; Modifies AC,XR, Clear I bit
GETRCVTIME   LDA #$E7
             STA PRTB        ;Set address to TEN output, (TEN DATA 0)
             SEI
             BCLR 5,PRTB     ;Enable transmitter latch
             CPX #0
             BEQ GETBIT0
             CPX #1 '
             BEQ GETBIT1
             CPX #2
             BEQ GETBITA2
             CPX #3
```

```
                BEQ GETBITA3    ;(Direct branch is out of range)
                CPX #4
                BEQ GETBITA4
                CPX #5
                BEQ GETBITA5
                CPX #6
                BEQ GETBITA6
                JMP GETBIT7

GETBITA2        JMP GETBIT2
GETBITA3        JMP GETBIT3
GETBITA4        JMP GETBIT4
GETBITA5        JMP GETBIT5
GETBITA6        JMP GETBIT6

GETBIT0         LDX #$D7        ;Code of start transmittion
                STX PRTB        ; 0.0uSec Start transmition
                CLRA            ; 1.5uS
                CLRA            ; 3.0uS
                CLRA            ; 4.5uS
                CLRA            ; 6.0uS
                CLRA            ; 7.5uS
                NOP             ; 8.5uS
                BRCLR 0,PRTA,WIDTHA100  ; 10.0uS (or less)
                BRCLR 0,PRTA,WIDTHA125  ; 12.5uS
                BRCLR 0,PRTA,WIDTHA150  ; 15.0uS
                BRCLR 0,PRTA,WIDTHA175  ; 17.5uS
                BRCLR 0,PRTA,WIDTHA200  ; 20.0uS
                BRCLR 0,PRTA,WIDTHA225  ; 22.5uS
                BRCLR 0,PRTA,WIDTHA250  ; 25.0uS
                BRCLR 0,PRTA,WIDTHA275  ; 27.5uS
                BRCLR 0,PRTA,WIDTHA300  ; 30.0uS
                BRCLR 0,PRTA,WIDTHA325  ; 32.5uS
                BRCLR 0,PRTA,WIDTHA350  ; 35.0uS
                BRCLR 0,PRTA,WIDTHA375  ; 37.5uS
                BRCLR 0,PRTA,WIDTHA400  ; 40.0uS
                BRCLR 0,PRTA,WIDTHA425  ; 42.5uS
                BRA WIDTHA450           ; 45.0uS (or more)

GETBIT1         LDX #$D7        ;Code of start transmittion
                STX PRTB        ; 0.0uSec Start transmition
                CLRA            ; 1.5uS
                CLRA            ; 3.0uS
                CLRA            ; 4.5uS
                CLRA            ; 6.0uS
                CLRA            ; 7.5uS
                NOP             ; 8.5uS
                BRCLR 1,PRTA,WIDTHA100  ; 10.0uS (or less)
                BRCLR 1,PRTA,WIDTHA125  ; 12.5uS
                BRCLR 1,PRTA,WIDTHA150  ; 15.0uS
                BRCLR 1,PRTA,WIDTHA175  ; 17.5uS
                BRCLR 1,PRTA,WIDTHA200  ; 20.0uS
                BRCLR 1,PRTA,WIDTHA225  ; 22.5uS
                BRCLR 1,PRTA,WIDTHA250  ; 25.0uS
                BRCLR 1,PRTA,WIDTHA275  ; 27.5uS
```

```
                BRCLR 1,PRTA,WIDTHA300  ; 30.0uS
                BRCLR 1,PRTA,WIDTHA325  ; 32.5uS
                BRCLR 1,PRTA,WIDTHA350  ; 35.0uS
                BRCLR 1,PRTA,WIDTHA375  ; 37.5uS
                BRCLR 1,PRTA,WIDTHA400  ; 40.0uS
                BRCLR 1,PRTA,WIDTHA425  ; 42.5uS
                BRA WIDTHA450           ; 45.0uS (or more)

WIDTHA450       INCA
WIDTHA425       INCA
WIDTHA400       INCA
WIDTHA375       INCA
WIDTHA350       INCA
WIDTHA325       INCA
WIDTHA300       INCA
WIDTHA275       INCA
WIDTHA250       INCA
WIDTHA225       INCA
WIDTHA200       INCA
WIDTHA175       INCA
WIDTHA150       INCA
WIDTHA125       INCA
WIDTHA100       BCLR 4,PRTB     ; Disable trasmit point
                BSET 5,PRTB     ; Disable transmit latch
                CLI             ; Enable interrupts again
                RTS

GETBIT2         LDX #$D7        ;Code of start transmittion
                STX PRTB        ; 0.0uSec Start transmition
                CLRA            ; 1.5uS
                CLRA            ; 3.0uS
                CLRA            ; 4.5uS
                CLRA            ; 6.0uS
                CLRA            ; 7.5uS
                NOP             ; 8.5uS
                BRCLR 2,PRTA,WIDTHA100  ; 10.0uS (or less)
                BRCLR 2,PRTA,WIDTHA125  ; 12.5uS
                BRCLR 2,PRTA,WIDTHA150  ; 15.0uS
                BRCLR 2,PRTA,WIDTHA175  ; 17.5uS
                BRCLR 2,PRTA,WIDTHA200  ; 20.0uS
                BRCLR 2,PRTA,WIDTHA225  ; 22.5uS
                BRCLR 2,PRTA,WIDTHA250  ; 25.0uS
                BRCLR 2,PRTA,WIDTHA275  ; 27.5uS
                BRCLR 2,PRTA,WIDTHA300  ; 30.0uS
                BRCLR 2,PRTA,WIDTHA325  ; 32.5uS
                BRCLR 2,PRTA,WIDTHA350  ; 35.0uS
                BRCLR 2,PRTA,WIDTHA375  ; 37.5uS
                BRCLR 2,PRTA,WIDTHA400  ; 40.0uS
                BRCLR 2,PRTA,WIDTHA425  ; 42.5uS
                BRA WIDTHA450           ; 45.0uS (or more)

GETBIT3         LDX #$D7        ;Code of start transmittion
                STX PRTB        ; 0.0uSec Start transmition
                CLRA            ; 1.5uS
                CLRA            ; 3.0uS
```

```
                CLRA            ; 4.5uS
                CLRA            ; 6.0uS
                CLRA            ; 7.5uS
                NOP             ; 8.5uS
                BRCLR 3,PRTA,WIDTHA100  ; 10.0uS (or less)
                BRCLR 3,PRTA,WIDTHA125  ; 12.5uS
                BRCLR 3,PRTA,WIDTHA150  ; 15.0uS
                BRCLR 3,PRTA,WIDTHA175  ; 17.5uS
                BRCLR 3,PRTA,WIDTHA200  ; 20.0uS
                BRCLR 3,PRTA,WIDTHA225  ; 22.5uS
                BRCLR 3,PRTA,WIDTHA250  ; 25.0uS
                BRCLR 3,PRTA,WIDTHA275  ; 27.5uS
                BRCLR 3,PRTA,WIDTHA300  ; 30.0uS
                BRCLR 3,PRTA,WIDTHA325  ; 32.5uS
                BRCLR 3,PRTA,WIDTHA350  ; 35.0uS
                BRCLR 3,PRTA,WIDTHA375  ; 37.5uS
                BRCLR 3,PRTA,WIDTHA400  ; 40.0uS
                BRCLR 3,PRTA,WIDTHA425  ; 42.5uS
                BRA WIDTHA450           ; 45.0uS (or more)

GETBIT4         LDX #$D7        ;Code of start transmittion
                STX PRTB        ; 0.0uSec Start transmition
                CLRA            ; 1.5uS
                CLRA            ; 3.0uS
                CLRA            ; 4.5uS
                CLRA            ; 6.0uS
                CLRA            ; 7.5uS
                NOP             ; 8.5uS
                BRCLR 4,PRTA,WIDTH100   ; 10.0uS (or less)
                BRCLR 4,PRTA,WIDTH125   ; 12.5uS
                BRCLR 4,PRTA,WIDTH150   ; 15.0uS
                BRCLR 4,PRTA,WIDTH175   ; 17.5uS
                BRCLR 4,PRTA,WIDTH200   ; 20.0uS
                BRCLR 4,PRTA,WIDTH225   ; 22.5uS
                BRCLR 4,PRTA,WIDTH250   ; 25.0uS
                BRCLR 4,PRTA,WIDTH275   ; 27.5uS
                BRCLR 4,PRTA,WIDTH300   ; 30.0uS
                BRCLR 4,PRTA,WIDTH325   ; 32.5uS
                BRCLR 4,PRTA,WIDTH350   ; 35.0uS
                BRCLR 4,PRTA,WIDTH375   ; 37.5uS
                BRCLR 4,PRTA,WIDTH400   ; 40.0uS
                BRCLR 4,PRTA,WIDTH425   ; 42.5uS
                BRA WIDTH450            ; 45.0uS (or more)

GETBIT5         LDX #$D7        ;Code of start transmittion
                STX PRTB        ; 0.0uSec Start transmition
                CLRA            ; 1.5uS
                CLRA            ; 3.0uS
                CLRA            ; 4.5uS
                CLRA            ; 6.0uS
                CLRA            ; 7.5uS
                NOP             ; 8.5uS
                BRCLR 5,PRTA,WIDTH100   ; 10.0uS (or less)
```

```
                    BRCLR 5,PRTA,WIDTH125  ; 12.5uS
                    BRCLR 5,PRTA,WIDTH150  ; 15.0uS
                    BRCLR 5,PRTA,WIDTH175  ; 17.5uS
                    BRCLR 5,PRTA,WIDTH200  ; 20.0uS
                    BRCLR 5,PRTA,WIDTH225  ; 22.5uS
                    BRCLR 5,PRTA,WIDTH250  ; 25.0uS
                    BRCLR 5,PRTA,WIDTH275  ; 27.5uS
                    BRCLR 5,PRTA,WIDTH300  ; 30.0uS
                    BRCLR 5,PRTA,WIDTH325  ; 32.5uS
                    BRCLR 5,PRTA,WIDTH350  ; 35.0uS
                    BRCLR 5,PRTA,WIDTH375  ; 37.5uS
                    BRCLR 5,PRTA,WIDTH400  ; 40.0uS
                    BRCLR 5,PRTA,WIDTH425  ; 42.5uS
                    BRA WIDTH450           ; 45.0uS (or more)

WIDTH450              INCA
WIDTH425              INCA
WIDTH400              INCA
WIDTH375              INCA
WIDTH350              INCA
WIDTH325              INCA
WIDTH300              INCA
WIDTH275              INCA
WIDTH250              INCA
WIDTH225              INCA
WIDTH200              INCA
WIDTH175              INCA
WIDTH150              INCA
WIDTH125              INCA
WIDTH100              BCLR 4,PRTB    ; Disable trasmit point
                    BSET 5,PRTB    ; Disable transmit latch
                    CLI            ; Enable interrupts again
                    RTS

GETBIT6             LDX #$D7       ;Code of start transmittion
                    STX PRTB       ; 0.0uSec Start transmition
                    CLRA           ; 1.5uS
                    CLRA           ; 3.0uS
                    CLRA           ; 4.5uS
                    CLRA           ; 6.0uS
                    CLRA           ; 7.5uS
                    NOP            ; 8.5uS
                    BRCLR 6,PRTA,WIDTH100  ; 10.0uS (or less)
                    BRCLR 6,PRTA,WIDTH125  ; 12.5uS
                    BRCLR 6,PRTA,WIDTH150  ; 15.0uS
                    BRCLR 6,PRTA,WIDTH175  ; 17.5uS
                    BRCLR 6,PRTA,WIDTH200  ; 20.0uS
                    BRCLR 6,PRTA,WIDTH225  ; 22.5uS
                    BRCLR 6,PRTA,WIDTH250  ; 25.0uS
                    BRCLR 6,PRTA,WIDTH275  ; 27.5uS
                    BRCLR 6,PRTA,WIDTH300  ; 30.0uS
                    BRCLR 6,PRTA,WIDTH325  ; 32.5uS
                    BRCLR 6,PRTA,WIDTH350  ; 35.0uS
                    BRCLR 6,PRTA,WIDTH375  ; 37.5uS
                    BRCLR 6,PRTA,WIDTH400  ; 40.0uS
```

```
                BRCLR 6,PRTA,WIDTH425  ; 42.5uS
                BRA WIDTH450           ; 45.0uS (or more)

GETBIT7         LDX #$D7       ;Code of start transmittion
                STX PRTB       ; 0.0uSec Start transmition
                CLRA           ; 1.5uS
                CLRA           ; 3.0uS
                CLRA           ; 4.5uS
                CLRA           ; 6.0uS
                CLRA           ; 7.5uS
                NOP            ; 8.5uS
                BRCLR 7,PRTA,WIDTH100  ; 10.0uS (or less)
                BRCLR 7,PRTA,WIDTH125  ; 12.5uS
                BRCLR 7,PRTA,WIDTH150  ; 15.0uS
                BRCLR 7,PRTA,WIDTH175  ; 17.5uS
                BRCLR 7,PRTA,WIDTH200  ; 20.0uS
                BRCLR 7,PRTA,WIDTH225  ; 22.5uS
                BRCLR 7,PRTA,WIDTH250  ; 25.0uS
                BRCLR 7,PRTA,WIDTH275  ; 27.5uS
                BRCLR 7,PRTA,WIDTH300  ; 30.0uS
                BRCLR 7,PRTA,WIDTH325  ; 32.5uS
                BRCLR 7,PRTA,WIDTH350  ; 35.0uS
                BRCLR 7,PRTA,WIDTH375  ; 37.5uS
                BRCLR 7,PRTA,WIDTH400  ; 40.0uS
                BRCLR 7,PRTA,WIDTH425  ; 42.5uS
                BRA WIDTH450           ; 45.0uS (or more)

****************************************
REPRXCARDS      LDA INSTBUF+3
                STA REPRXADDR   ;Start loop from couple cards INSTBUF+3
                LDA INSTBUF+4
                STA TOREPRXADDR ;to cards INSTBUF+4
                BSET 6,CMDCODE  ;Ready to recieve next command
REPRXCARDSLP    LDA REPRXADDR
                ORA #$F0
                SEI
                STA PRTB        ;Store REN address
                BSET 0,PRTC     ;Enable REN latch for writing REN address
                NOP
                BCLR 0,PRTC     ;Disable REN latch
                CLI
                JSR REP2RXCARD
                LDA #$80
                JSR SCFAPPEND   ;Start transmittion of new cards connectivity
                LDA ID
                JSR SCFAPPEND   ;transmit ID
                LDA #11         ;Command code
                JSR SCFAPPEND
                LDA REPRXADDR   ;RX enable address
                JSR SCFAPPEND
                LDA BYRCV       ;RX present status
                JSR SCFAPPEND
                TST BYRCV       ;Is non of 2 cards present?
                BEQ REPRXNEXT   ;if not present continue to next 2 cards
                LDX #0
```

```
REPRXSCLOOP  LDA PYRCV,x      ; Loop to send all column core connectivity
             JSR SCFAPPEND
             INCX             ; Next column
             CPX #8           ; last column?
             BNE REPRXSCLOOP  ; No - continue to next
REPRXNEXT    LDA REPRXADDR
             CMP TOREPRXADDR  ; REPRXADDR = TOREPRXADDR ?
             BEQ REPRXEND     ; if yes last card reported
             LDA #1
             JSR SCFAPPEND    ; 1 means more cards to transmit
             LDA CMDCODE
             CMP #$80         ; Recieved stop command?
             BEQ REPRXEND     ; if yes, stop reporting scanning
             INC REPRXADDR    ; Next couple cards enable
             BRA REPRXCARDSLP
REPRXEND     LDA #0
             JSR SCFAPPEND    ; 0 means end of message
             RTS

*****************************
; Report couple reciver cards connectivity (no dry contact)
; Set the address of REN# before using this subroutine
; it check if card connected by reading RL (PRTA) before
; any transmittion/selecting and if PRTA=$FF no card present,
; $00 both cards present and $0F or $F0 is one of cards connected,
; (other values indicate noise or hardware failure)
; If the card present check connectivity of it's point by selecting
; column of receiving (turning the column NPN on), wait RXREL time
; and than turning the NPN off, after turning off each connected point
; will read 1 (PRTA) for many uSec because of increasing voltage in
; input of RCVAMP (see schematics)
; it returns:
; BYRCV = NOT RL[0..7] before any YRCV select
; if equal $00 no card of couple present and don't continue
; to calculate PYRCV+[0..7]
; PYRCV+x = a byte of data of PRTA*BYRCV after turn off YRCV select
; it should read 1 if and only if point connected to core and card present
; x equals the column number of selecting
; Modifies AC,XR,TMP1,TMP2
REP2RXCARD   LDX #7
REPRXCLR     CLR PYRCV,x      ;Clear PYRCV (No coil connected)
             DECX
             BPL REPRXCLR     ; also gives short delay
             JSR GETPRTA      ; Get PRTA silently, set I-flag
             CLI              ; Clear I-flag
             COMA             ;AC = NOT AC (for each bit)
             STA BYRCV
             BEQ REPRXRTS     ;No card present, return
             LDX #0           ;Start yrcv loop from column 0
             STX REPRXCOL
REPRXYLOOP   LDX REPRXCOL     ;XR = Column to select
             JSR TWOPOWERX    ;AC = 2^XR
             LDX #2           ;YRCV latch write
             JSR LATCHWR      ;Turn column x off YRCV on (effect TMP1,TMP2)
             GO_TIMER RXREL
```

```
                    WAIT_TIMER      ;Wait RXREL time for RCVAMP to relax
                    LDA REPRXCOL
                    ORA #$E0
                    STA PRTB
                    SEI
                    BCLR 7,PRTB
                    NOP
                    BSET 7,PRTB
                    NOP
                    NOP
                    NOP
                    NOP
                    NOP
                    LDA PRTA        ;About 9.0 uSec after turning NPN off
                    NOP
                    NOP
                    NOP
                    NOP
                    NOP
                    NOP
                    LDX REPRXCOL
                    AND PRTA        ;Again after 18.0uSec To prevent effect of narrow noise
pulse
                    AND BYRCV
                    STA PYRCV,x
                    CLI
                    INC REPRXCOL
                    LDX REPRXCOL
                    CPX #8
                    BNE REPRXYLOOP
                    GO_TIMER RXREL
                    WAIT_TIMER      ;Wait to ensure relax of RCVAMP before other actions
          REPRXRTS  RTS

          ************************
          ; Transmit at sepcified point repeatedly (pulses),
          ; Useful for osciloscope check (Use TEN for trigger)
          ; Command format from RS-232:
          ;  (idle) ,ID,9,7,RX y-rcv select latch,
          ;         TX y-driver latch, transmitter address (TA),TON,TOFF
          ;  TON - time transmitter on (on time = 3uSec*TON+4uSec), (0 counts 256)
          ; TOFF - time transmitter off (off time = 3uSec*TOFF+13.5uSec) (0 counts 256)
          ; The scanner will repeatedly transmit at the specified point
          ;     until recieved it's ID or 128 from the terminal
          ; When start sending pulses it will send terminal:
          ;      128,ID,9,1,0
          ; When stopped: 128,ID,9,0
          ; Note that the interupts may occure while sending pulses, so
          ; sometimes a pulse will be longer than allowed
          ; WARNING: DUTY CYCLE SHOULD NOT BE OVER 40% BECAUSE OF POWER DISPATION
          ;          OF TRANSMITTER RESISTOR!
          TESTPOINT   LDX #2          ; Choose Transmitter y-driver latch
                      LDA INSTBUF+3   ; Y-rcv select latch data
                      JSR LATCHWR     ; Write it to latch
                      LDX #1          ; Choose Transmitter y-driver latch
                      LDA INSTBUF+4   ; Y-drv latch data
```

```
                JSR LATCHWR     ; Write it to latch
                LDX #0          ; Choose transmitter latch
                LDA INSTBUF+5   ; Transmitter latch data (TA)
                JSR LATCHWR     ; Choose transmitter
                CLR INSTBUF     ; Reset instruction buffer
                LDA #$80        ; Send 128 at start of message
                JSR SCFAPPEND   ; Send it
                LDA ID
                JSR SCFAPPEND   ; Send ID
                LDA #9
                JSR SCFAPPEND   ; Send command code
                LDA #0
                JSR SCFAPPEND   ; Send end of message (means start TESTPOINT)
                LDA #$E7
                STA PRTB        ;Set address to TEN output, DATA 0
                BCLR 5,PRTB     ;Enable writing to transmitter latch
                BCLR NDSR,PRTC  ;Ready to recieve data from RS-232 (DSR)
TESTPLOOP       BSET 4,PRTB     ;Transmitter Enable (TEN)
                LDX INSTBUF+6
TESTPDEL1       DECX             ;Delay before stop tranmitter
                BNE TESTPDEL1
                BCLR 4,PRTB
                LDX INSTBUF+7
TESTPDEL2       DECX
                BNE TESTPDEL2
                LDA SCITA
                CMP SCITEA       ; Test if anything to transmit
                BEQ TESTPNSEND   ; If SCBUF empty continue
                BRSET NDTR,PRTC,TESTPNSEND ;Check if Data Set Ready
                JSR SENDDATA     ;Sending data if terminal ready
                                 ;to start interputs of transmitter
TESTPNSEND      BRCLR RDRF,SCSR,TESTPLOOP  ; Continue if didn't received anything
                                 ; From RS-232
                LDA SCDR         ; Check data recieved
                CMP ID           ; Is data recieved for this scanner ID
                BEQ TESTPSTOP    ; If yes end of TESTPOINT
                CMP #$80         ; Or if recieved for all scanners
                BEQ TESTPSTOP
                BSET RWU,SCCR2   ; If not sleep SC reciever until line idle
                BRA TESTPDEL2
TESTPSTOP       BSET 6,INSTBUF+1 ; Mark that ready to recieve next instruction
                BSET NDSR,PRTC   ; RS-232 not DSR
                LDA #$80         ; Send 128 at start of message
                JSR SCFAPPEND    ; Send it
                LDA ID
                JSR SCFAPPEND    ; Send ID
                LDA #9
                JSR SCFAPPEND    ; Send command code
                LDA #1
                JSR SCFAPPEND    ; Send test point over
                LDA #0
                JSR SCFAPPEND    ; Send end of message
                RTS
```

```
*****************************************
; Latches initialization
; (TX=00, Y-SELECT#,Y-DRIVER#=FF)
; Modifies TMP1,TMP2,AX,XR
LAINIT:     CLRA
            CLRX
            JSR LATCHWR  ; TX Disable (latch 0)
            CLRA
            LDX #2
            JSR LATCHWR  ; RX Y-SELECT OFF (latch 2)
            CLRA
            LDX #1
            JSR LATCHWR  ; TX Y-DRIVER OFF (latch 1)
            RTS

***********************
; READ ID OF MCU (serial read of PD7 to ID)
; Modify AC,XR
GETID:      LDX #$F7
IDLOOP1:    STX PRTB
            LDA PRTD
            LSLA
            ROL ID
            DECX
            CPX #$EF
            BNE IDLOOP1
            RTS

;*****************************************
; Function for writing data to latch
; AC = data
; XR = 0 for transmiter latch
;      1 for Ydriver latch
;      2 for Yreciever select latch
;      3 (or more) for I/O unit latch
; modify TMP1=0,TMP2

LATCHWR:    STA TMP1
            STA TMP2
            LDA #$07
LALOOP1:    LSL TMP1
            BCC LADT0
            ORA #$10      ; data is 1
LADT0:      ORA #$E0      ; all write disabled
            STA PRTB
            TSTX
            BNE LA1
            BCLR 5,PRTB   ; WR0  (TA[0..6],TE)
            BRA LAWD
LA1         CPX #1
            BNE LA2
            BCLR 6,PRTB   ; WR1  (YDRV#[0..7])
            BRA LAWD
LA2:        CPX #2
            BNE LA3
```

```
                BCLR 7,PRTB    ; WR2  (YRCV#[0..7])
                BRA LAWD
LA3:            BSET 1,PRTC    ; WR0  (I/O CARD LATCH)
LAWD:           STA PRTB       ; Write Disable all controler latches
                BCLR 1,PRTC    ; Write Disable I/O latch
                AND #$07
                DECA
                BPL LALOOP1
                LDA TMP2
                RTS

;**********************
;AC=2^XR
;Modifies AC,XR
TWOPOWERX       LDA #1
TWOPOWLP        DECX
                BMI TWOPOWRTS
                LSLA     ;AC=AC*2
                BRA TWOPOWLP
TWOPOWRTS       RTS


;**************************************
;TIMER PROCEDURES
;**************************************

;******************
;TIMER interupt
QTIMER:         RTI             ; TMP

;******************
;Next Transmit TIMER
;Until TOCF bit is set don't transmit again,
;used to ensure relaxing of oscilation from previos
;transmit or column select in reciever!
;Time set to (AC*256+RX)*(2uSEC)  (Must be > 20uSEC)
;(if previous time defined not over yet,
; define the new time - assuming it is longer.)
;Modifies AC,RX,TMP1
NTTIMER:    ADD TACH        ;Start count from alternate counter
            STA TMP1        ;Timer hi
            TXA
            ADD TACL
            BCC NTTOK1
            INC TMP1        ;Add carry to timer hi
NTTOK1:     TAX
            LDA TMP1
            STA TOCMPH
            LDA TSR         ;Read status register to clear it
            STX TOCMPL
            RTS

; Serial communication
```

```
**********
; Serial Communication Initialization
; including: set buad rate,interupt enable,reset buffer
; baud rate:9600, M flag set to 9 data bits (8 data + 1 parity)
; Send zero byte to terminal
; Modifies AC
SCINIT      LDA #$04        ; Disable all SCI interupts,TE
                            ; and Enable reciever (RE)
            STA SCCR2       ; (Transmitter disabled)
            BSET NDSR,PRTC  ; Not DSR, RS-232 not ready - yet
            BSET M,SCCR1    ; 8 bit data + 1 bit parity
            BCLR WAKE,SCCR1 ; Wake up method is idle line detect
            LDA #$30
            STA SCBRR       ; set buad rate to 9600Hz (4.0MHz crystal ?)
            CLR SCITA       ; Reset SCI buffer
            CLR SCITEA
            CLR INSTBUF
            LDA #$40
            STA CMDCODE     ; Ready to recieve instruction
            RTS

**************************
; Store I flag and set it
STOREI      BIL OKI1
            BCLR 0,TSTORE
            SEI
            RTS
OKI1        BSET 0,TSTORE
            RTS

**************
; Clear I flag after STOREI if it was high
RESTOREI    BRSET 0,TSTORE,OKI2
            CLI
OKI2        RTS

**************************
; Serial Communication Interupt
QSCI        JSR SENDDATA
            RTI

************************************
; SC Change to data transmit mode
; Change ONLY IF SCBUF NOT EMPTY!
; turn of RS-232 DSR, Disable reciver, enable transmitter and
; send first data from buffer
; NOTE: TO START TRANSMIT FILL (1 byte at least) SCBUF BEFORE USING THIS ROUTINE
; Modifies AC,XR,QTMP1

**************************
; append data byte to SC transmit buffer
; data passed by AC
; it also try to send data if possible
; SET I FLAG BEFORE CALLING THIS SUBROUTINE TO DISABLE INTERUPTS
; Modifies TMP1 (=XR), TMP2 (=AC)
```

```
; set carry if no space for more data (no send)
SCAPDATA    STX TMP1
            STA TMP2
            JSR SCFINDFREE  ;Check if buffer full
            BEQ SCNOFREE
            LDA TMP2        ;Restore data to accumulator
            LDX SCITEA      ;load transmitter data end of buffer point
            STA SCBUF,x     ;Store data in buffer
            INCX
            CPX #SCBUFSIZE
            BNE SCAPOK1
            CLRX
SCAPOK1     STX SCITEA      ;set new end of buffer address
            JSR SENDDATA    ;This send data if possible (Terminal Ready,
                            ; Transmit register free etc.)
                            ; This also activate interupt if first transmition
            CLC
            LDA TMP2
            LDX TMP1
            RTS
SCNOFREE    JSR SENDDATA    ;Send data if possible
            LDA TMP2        ;Restore registers
            LDX TMP1
            SEC             ;Carry indicates no free space in buffer for data
            RTS

*****************************
; append data byte to SC by using SCAPDATA (look above)
; but here if buffer full
; retry to append until success or recieved STOP command from terminal
; (SC Forced appending data)
; Data passed by accumulator
; Modifies TMP1(=XR) ,TMP2(=AC)
SCFAPPEND   JSR STOREI      ; Prevent interupts during send data
            STX TMP1        ; Store XR    TMPORARY!!!
            LDX CMDCODE     ; Load command in instruction buffer
            CPX #$80        ; Compare to stop command
            BEQ SCFAPRTS    ; Recieved STOP command, Return without transmit
            LDX TMP1
RSCFAP      SEI
            JSR SCAPDATA    ; Try appending data and store XR in TMP1
            BCC SCFAPRTS    ; Succeed, (SC transmit buffer not full)
            CLI
            BRCLR NDTR,PRTC,RETRYAP ; If terminal can't recieve try
            JSR READDATA    ; recieving instruction from terminal
            LDX INSTBUF+1   ; Load command in instruction buffer
            CPX #$80        ; Compare to stop command
            BEQ SCFAPRTS    ; Recieved STOP command, Return
RETRYAP     LDX TMP1        ; Restore XR
            BRA RSCFAP      ; Next try to transmit
SCFAPRTS    LDX TMP1        ; Restore XR
            JMP RESTOREI    ; Restore value of I flag

**************
;Find (calculate) free space in SC buffer
```

```
;(return answer in AC)
;Modifies AC, Z=1 If place available
SCFINDFREE   LDA SCITA
             SUB SCITEA
             BHI SCPLUS
             ADD #{ SCBUFSIZE-1 }
             RTS
SCPLUS       SUB #1
             RTS

*********************
; send data to rs232 sci, use 9th data bit as even parity bit
; If transmit data register full
; suspend sending data until next call to this subroutine
; If SCBUF empty or DTR, change to receive mode
; SET I FLAG BEFORE CALLING THIS SUBROUTINE TO DISABLE INTERUPTS
; Modifies AC,XR,QTMP1
SENDDATA     BRSET TC,SCSR,SCTC     ; If transmit completed skip check of
                                    ; transmit register
             BRCLR TDRE,SCSR,SDRTS ; Transmit register not empty, no sending
SCTC         LDX SCITA         ; load current transmit offset address
             CPX SCITEA        ; check if reached end address
             BEQ SCTDIS        ; If end goto transmit disable (end of transmition)
             BRCLR NDTR,PRTC,OKSEND  ; if terminal ready goto OKSEND
SCTDIS       BCLR TIE,SCCR2    ; No more data in buffer to transmit:
                               ; Disable transmitter interupts
             BCLR TE,SCCR2     ; Disable transmitter
             RTS
OKSEND       LDA SCBUF,x       ; load data to send
             STA SCDR          ; write data to SC data register
             JSR CHECKPARITY
             BCS SETPARITY     ; ODD
             BCLR 6,SCCR1      ; clear even parity bit
             BRA SENDOK1
SETPARITY    BSET 6,SCCR1      ; set even parity bit
SENDOK1      BSET TE,SCCR2      ; SC Transmitter Enable
             BSET TIE,SCCR2     ; Enable SC transmit interupt
             LDX SCITA
             CPX #{ SCBUFSIZE-1 }     ;check if reached end of buffer
             BNE SENDOK2
             LDX #$FF          ; reached buffer end address
                               ; set address to start of buffer
SENDOK2      INCX
             STX SCITA         ; set to next address to send
SDRTS        RTS

****************************
;Recieve data from terminal (not interupt)
;If after DSR line steel idle, return
;Else recieve all message or return if message regarding
; other ID
; Modifies AC,XR,TMP1
READDATA     BRSET 6,CMDCODE,OKREAD  ; check if ready to recieve command
READDATARTS  RTS               ; Not ready to recieve next command
OKREAD       LDA SCSR
```

```
                AND #%00110000    ; Check only RDRF and IDLE
                CMP #%00100000    ; If data recieved and line not idle yet
                BEQ READDATARTS   ; Ignore message and wait for idle line
                LDA SCDR          ; Access to data register to clear status register
                BCLR NDSR,PRTC    ; Data Set Ready - inform RS-232 terminal to send
                CLR INSTBUF       ; Reset instruction buffer
                LDA #$FF
                STA INSTBUF+2     ; Mark length of message
                LDA #$40
                STA CMDCODE
WAITDATA        CLRX
WAITDT1         BRSET RDRF,SCSR,DATARCV      ; Loop wait about 7 mSec for data
                BRSET IDLE,SCSR,CHECKLEN     ; exit loop: if line idle (end of message
                                             ;            message or recieved data)
                LDA #$08
WAITLOOPDT      SUB #1
                BNE WAITLOOPDT     ; Delay
                DECX
                BNE WAITDT1
                JMP CHECKLEN       ; No new data after delay, Check length of message
DATARCV         BRSET 1,SCSR,SCERROR        ; Framing Error
                BRSET 3,SCSR,SCERROR        ; Overrun Error
                LDA SCDR           ; Read Recieved Data
                LDX INSTBUF        ; Read Offset to store new data
                CPX #INSTBUFSIZE
                BEQ SCERROR        ; Not enough place in buffer for new data
                STA TMP1         ; Store data in temporary ram
                JSR CHECKPARITY    ; Set Carry if even parity should be 1
                BCS RCVODD
                BRSET 7,SCCR1,SCERROR    ; Parity error
                BRA RCVDATAOK1
RCVODD          BRCLR 7,SCCR1,SCERROR    ; Parity error
RCVDATAOK1      LDX INSTBUF
                BNE RCVDAT1        ; Check if first byte of data in message
                ; First byte is ID of card
                BRSET 7,TMP1,SCIDENT    ; Instruction regarding all cards
                LDA ID             ; Check if regarding this scanner (by ID)
                AND #$3F
                CMP TMP1
                BEQ SCIDENT        ; OK, continue recieve rest of message
                BSET RWU,SCCR2     ; Reciever go sleep until next time line is idle
                JMP ENDDATARCV     ; Data recieved not regarding this scanner
                                   ; Ignore message
SCERROR         BSET 5,INSTBUF+1 ; Set error bit
                BSET RWU,SCCR2     ; Reciever go sleep until next time line is idle
                JMP ENDDATARCV     ; Ignore rest of message
SCIDENT         LDA #1
                STA INSTBUF
                JMP WAITDATA       ; Continue read data
RCVDAT1         LDA TMP1         ; Load data to AC
                STA INSTBUF,x      ; Store data to instruction buffer
                INCX
                STX INSTBUF        ; Points next place to put data
                DECX
                CPX #1
```

```
                BLS WAITDATA      ; OK. Continue recieving data
                CPX INSTBUF+2
                BHI SCERROR       ; Message to long, ignore it
                BNE WAITDATA      ; Continue read data
                BSET 7,INSTBUF+1  ; Finished reading all data of this instruction
                JMP ENDDATARCV
CHECKLEN        LDX INSTBUF       ; Check lenght of message
                DECX
                CPX INSTBUF+2
                BEQ ENDDATARCV    ; message length ok
                BSET 5,INSTBUF+1  ; Error - message length not match
ENDDATARCV      BSET NDSR,PRTC    ; Not DSR (RS-232 Data Set Ready = 0)
DATAREADRTS     RTS

*************
; Check Parity bit of accumulator
; set carry if odd, clear otherwise
; Modifies AC,XR
CHECKPARITY     LDX #$09
PEVEN           DECX              ; calculating parity bit
                BEQ PARITYEVEN    ; all bits checked
                LSRA
                BCC PEVEN
PODD            DECX
                BEQ PARITYODD
                LSRA
                BCS PEVEN
                BRA PODD
PARITYODD       SEC
                RTS
PARITYEVEN      CLC
                RTS
```

Claims

1. Apparatus for providing an indication of the connection pattern of a multiplicity of data ports, pluralities of which are interconnected by conductors, said apparatus comprising:

signal transducer means operatively associated with at least some of said conductors at the ends thereof adjacent said data ports, at least one of said signal transducer means associated with at least one of said conductors being operative to impose a signal on a portion of said conductor and at least one of said signal transducer means associated with at least one of said conductors being operative to pick off said signal from said conductor;

means, connected to said transducer means, for identifying the existence of signal paths along said conductors between said pluralities of ports; and

output means, coupled to said means for identifying, for providing an output indication of said connection pattern produced by connection of said conductors to said pluralities of ports.

2. Apparatus according to claim 1 and also comprising means for distinguishing signals passing along predetermined signal paths from signals passing along other signal paths, according to their phase.

3. Apparatus according to claim 1 or claim 2 and also comprising means for distinguishing signals passing along predetermined signal paths from signals passing along other signal paths, according to their delay time constant.

4. Apparatus according to any of claims 1 - 3 and also comprising means for distinguishing signals passing along predetermined signal paths from signals passing along other signal paths, according to their amplitude.

5. Apparatus according to any of claims 1 - 4 wherein at least some of said conductors are arranged in a plurality of cables, each cable including at least one of said conductors.

6. Apparatus according to claim 5 wherein the plurality of cables comprise shielded cables and wherein said signal transducer means comprises induction means operatively associated with shielding of said shielded cables at the ends thereof adjacent said data ports, at least one of said induction means associated with each shielded cable being operative to impose a signal on said shielding of said cable and at least one of said induction means associated with each shielded cable being operative to pick off said signal from said shielding of said cable.

7. Apparatus according to any of claims 1 - 6 wherein said signal transducer means is operative to impose a signal on at least one conductor which does not carry any other signal.

8. Apparatus according to any of claims 1 - 7 wherein said signal transducer means is operative to impose a signal on at least one conductor which may carry other signals and includes means for isolating said signal imposed thereby from said other signals, thereby to prevent unacceptable interference therewith.

9. Apparatus according to any of claims 1 - 8 wherein said indication of said connection pattern of said data ports is provided automatically.

10. Apparatus according to any of the preceding claims and also comprising visual indicators associated with each of said multiplicity of data ports and apparatus for simultaneously operating the visual indicators associated with interconnected data ports, thereby to provide a visible indication of the interconnection therebetween.

11. Apparatus according to claim 10 and wherein said visual indicators are LEDs electrically associated with said induction means.

12. Apparatus according to any of the preceding claims and also comprising light source apparatus associated with each of said ports for providing a visible indication of pairs of interconnected ports.

13. Apparatus according to claim 12 and also comprising manually controllable means for scanning said ports to provide indication of said pairs of interconnected ports by said light source apparatus.

14. A local area network comprising cabling interconnecting a plurality of workstations, said cabling comprising apparatus according to any of claims 1 - 13 wherein said conductors are used for selectable and removable interconnection between selected ones of said data ports.

15. A computer system comprising at least one main computer, a plurality of workstations and a local area network interconnecting said at least one main computer and said plurality of workstations, said local area network comprising:

apparatus according to any of claims 1 - 13 wherein the multiplicity of data ports includes at least one computer port and a plurality of user ports and wherein said conductors are used for selectable and removable interconnection between selected ones of said user ports and said at least one computer port, thereby providing an indication of the connection pattern of said at least one computer port and said user ports.

FIG.1A

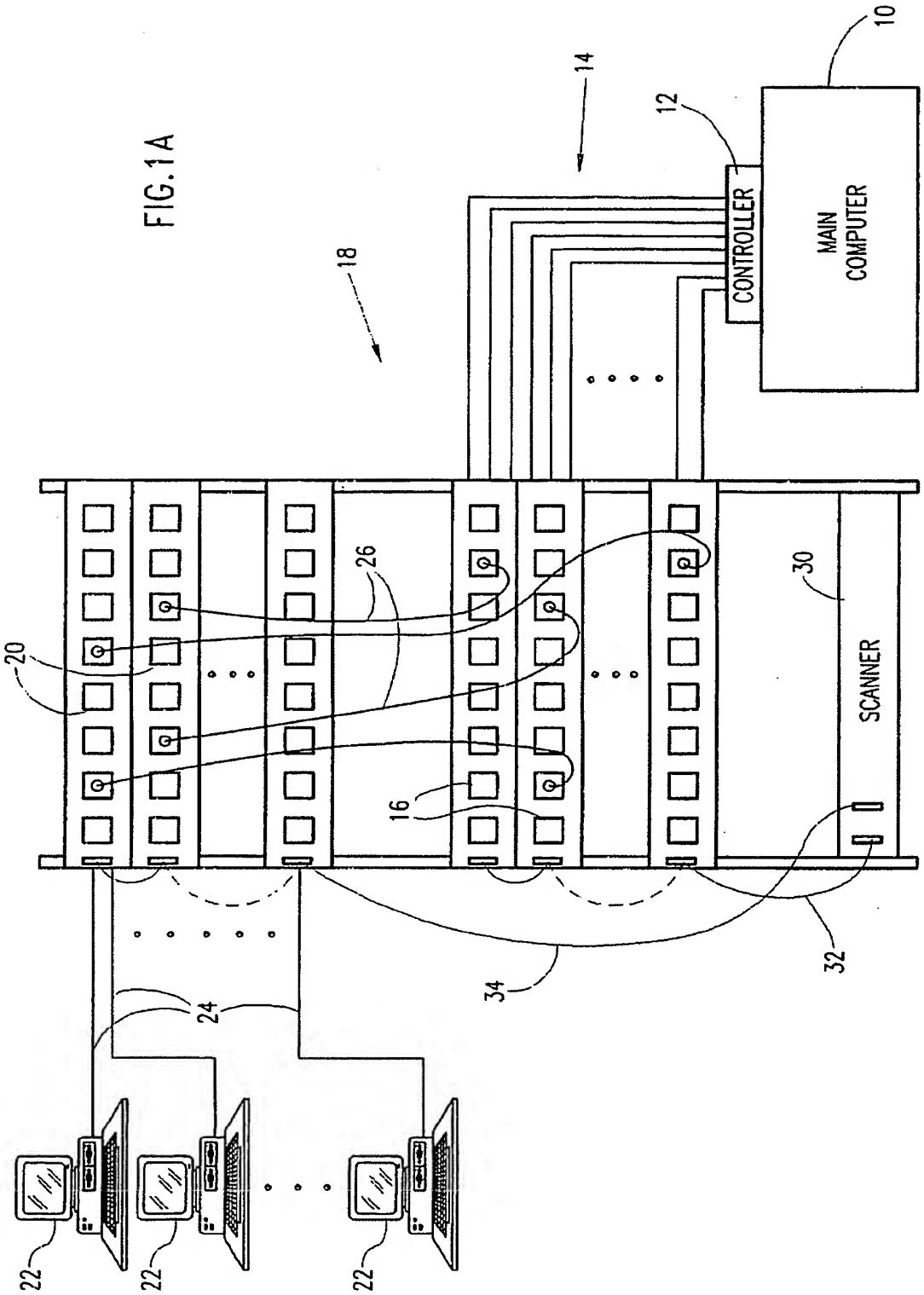
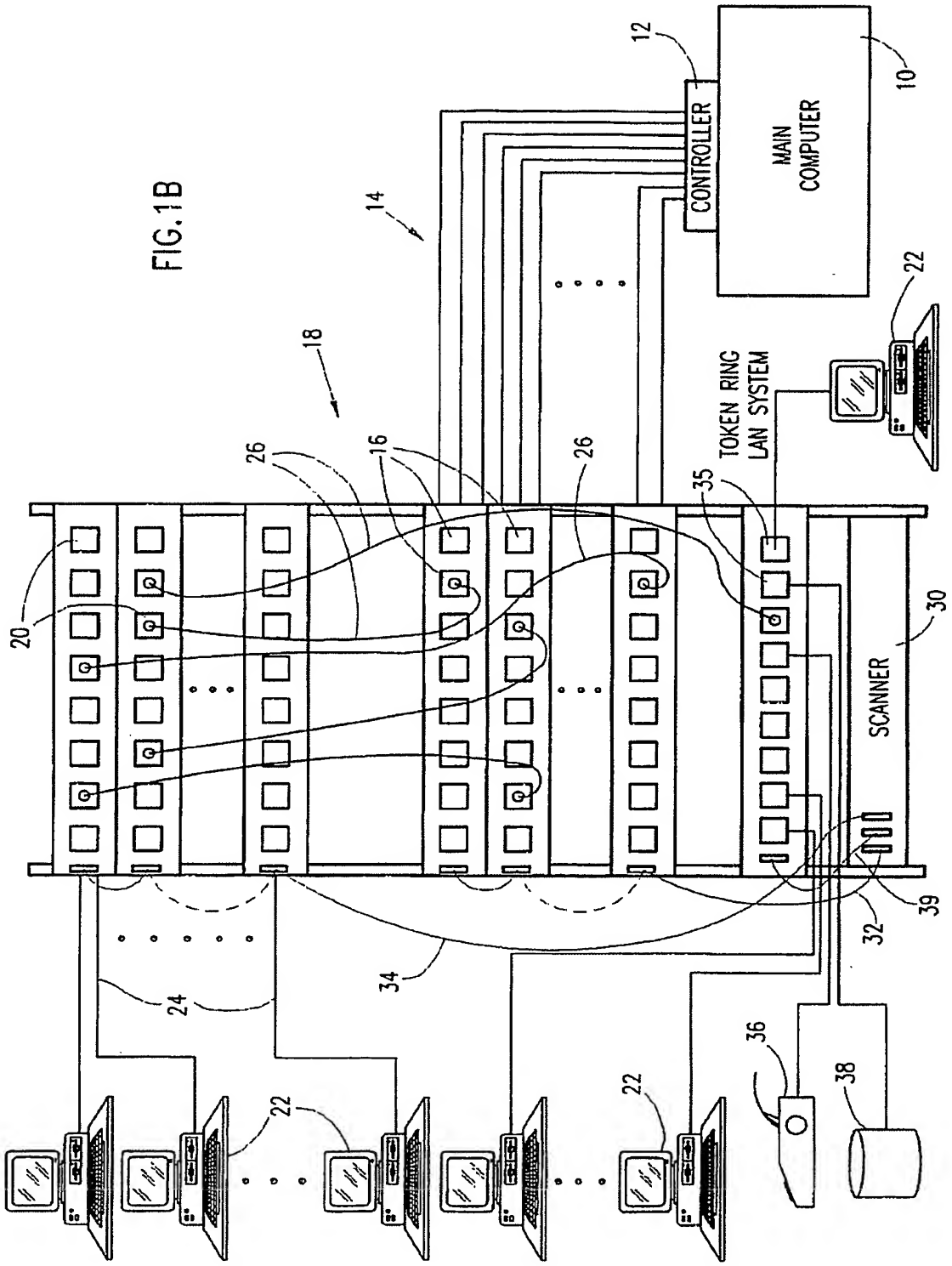

10
14
12
CONTROLLER
MAIN COMPUTER
18
26
30
20
SCANNER
16
32
34
24
22
22
22

FIG.1B


12
CONTROLLER
MAIN COMPUTER
10
14
18
22
TOKEN RING LAN SYSTEM
26
16
26
35
20
30
SCANNER
39
32
34
24
36
38
22
22

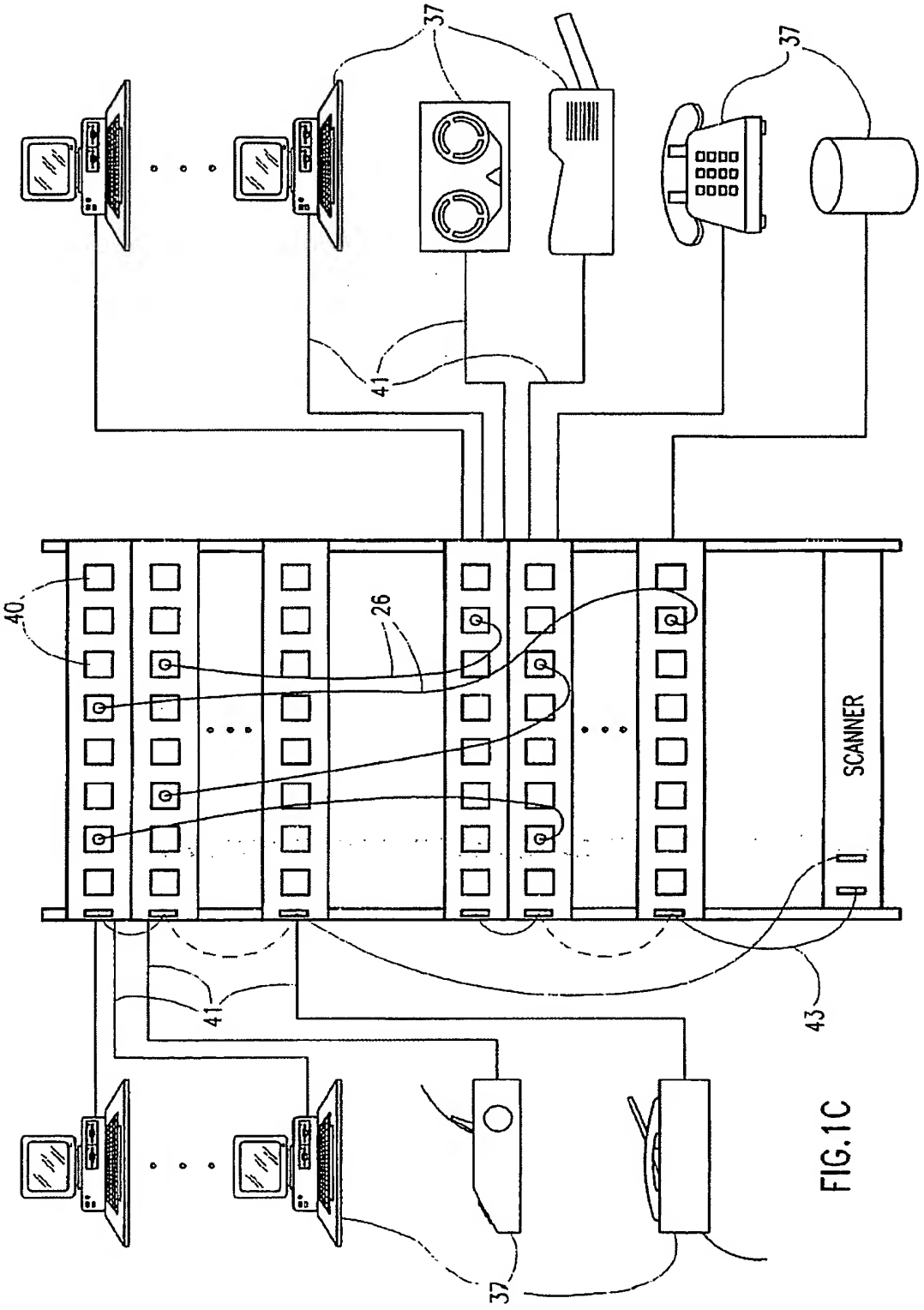


FIG.1C

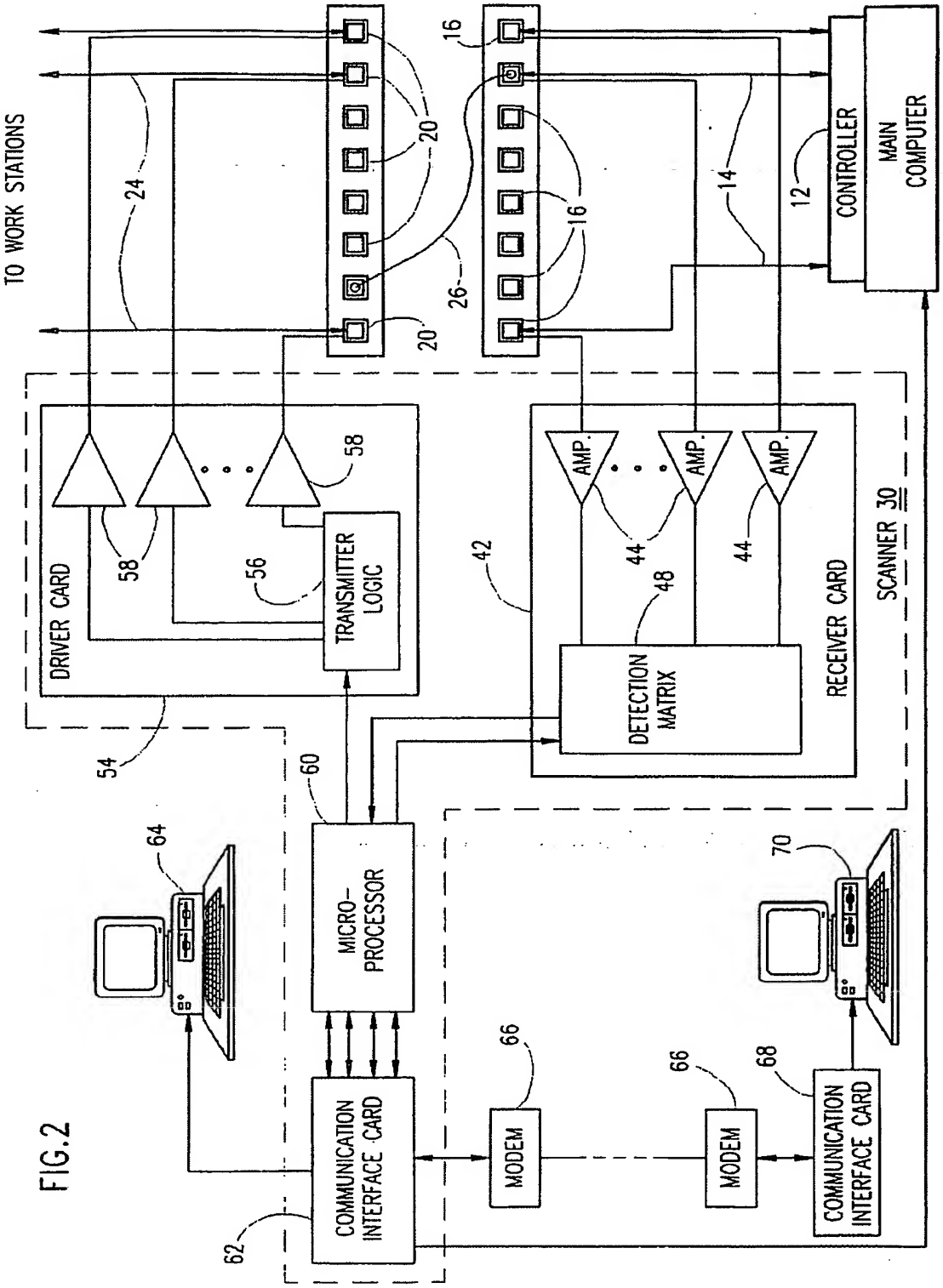

TO WORK STATIONS
24
20
20
16
16
26
14
12
CONTROLLER
MAIN COMPUTER
SCANNER 30
DRIVER CARD
58
58
56
TRANSMITTER LOGIC
42
AMP.
AMP.
AMP.
44
44
48
DETECTION MATRIX
RECEIVER CARD
54
60
64
MICRO-PROCESSOR
70
66
66
68
COMMUNICATION INTERFACE CARD
MODEM
MODEM
COMMUNICATION INTERFACE CARD
62

FIG.2

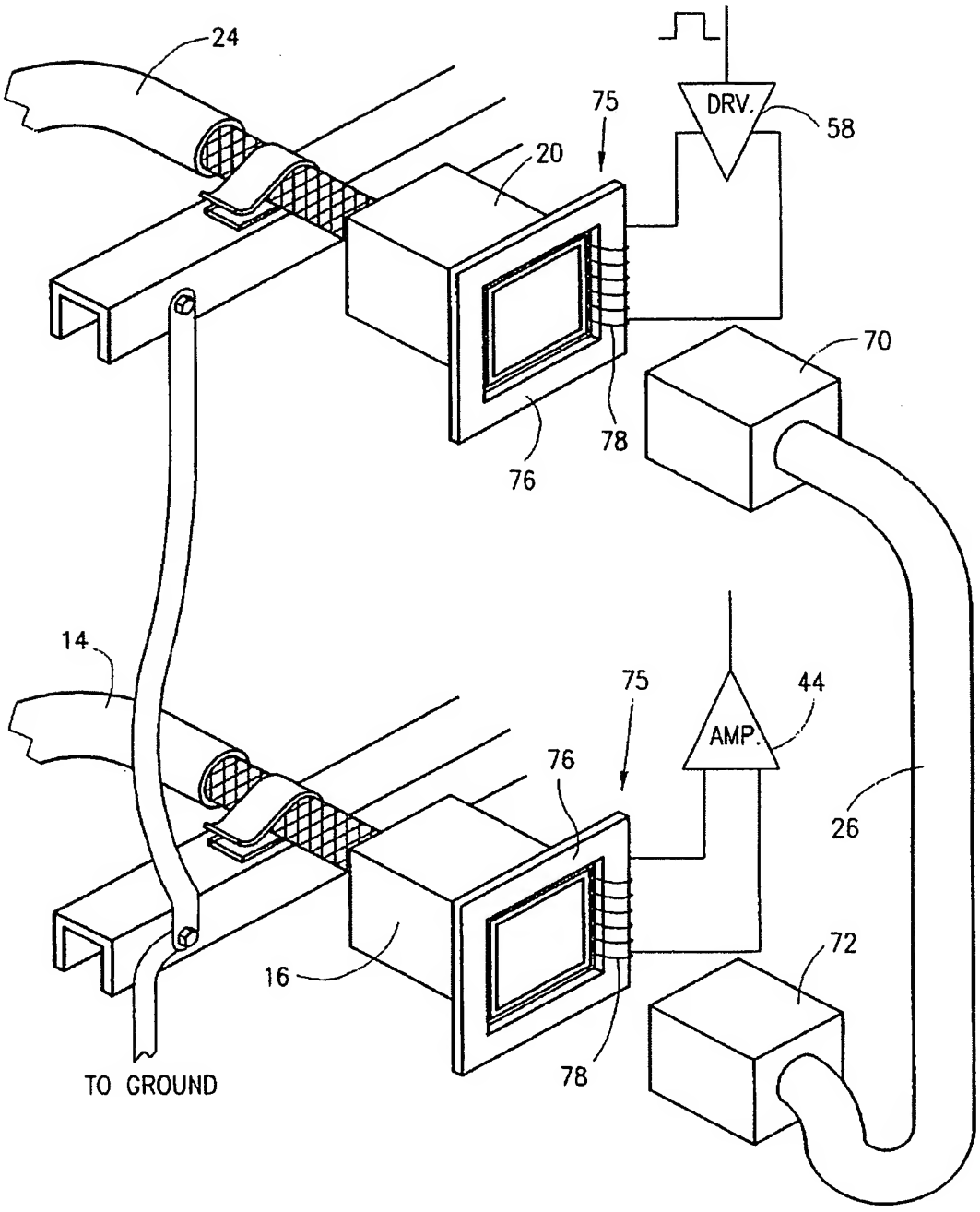


FIG.3

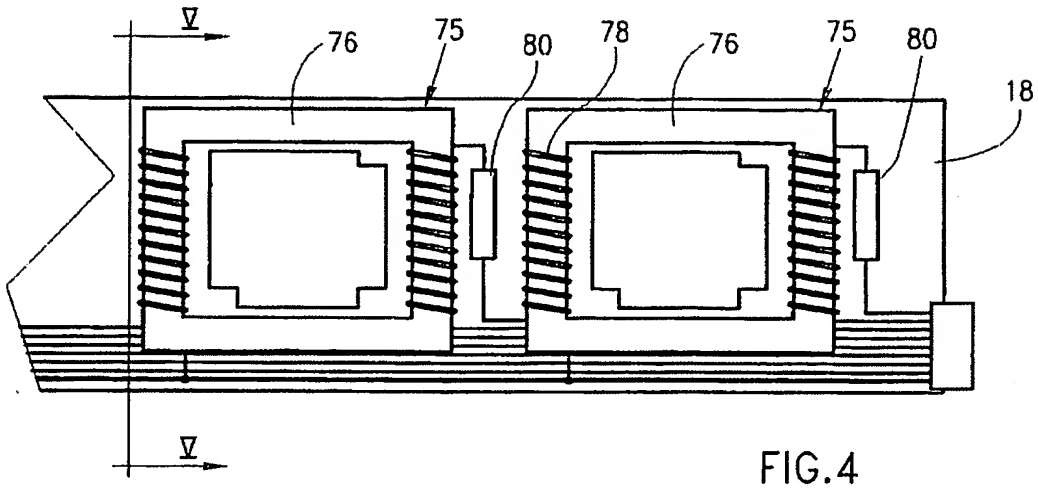


FIG.4

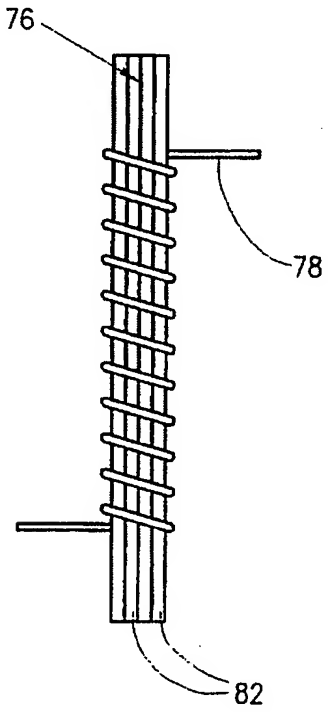


FIG.5

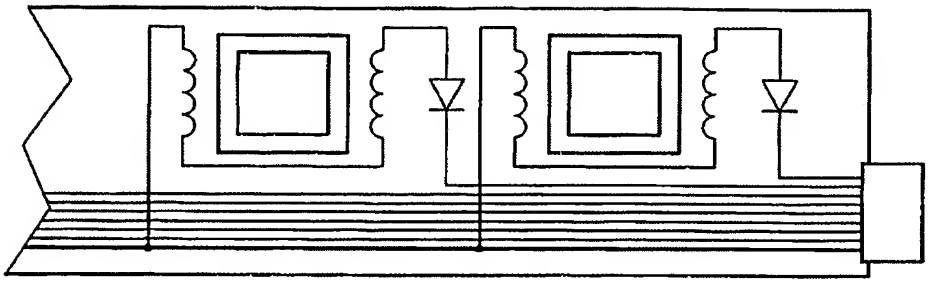


FIG.6A

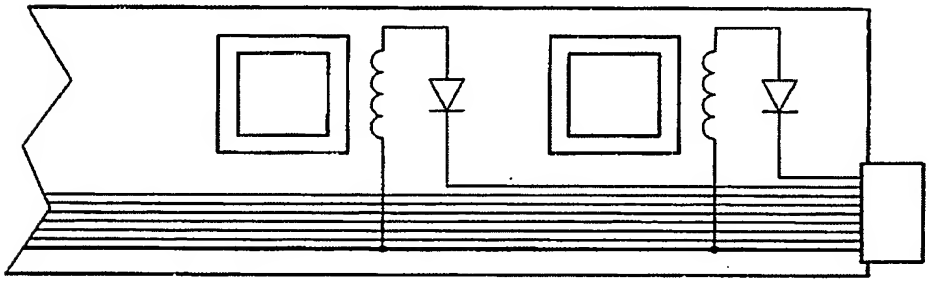


FIG.6B

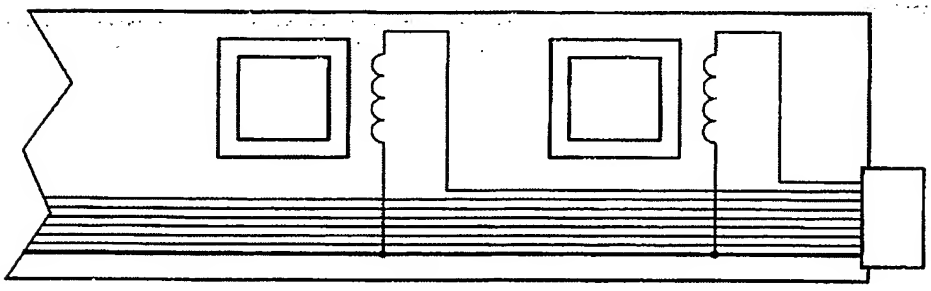


FIG.6C

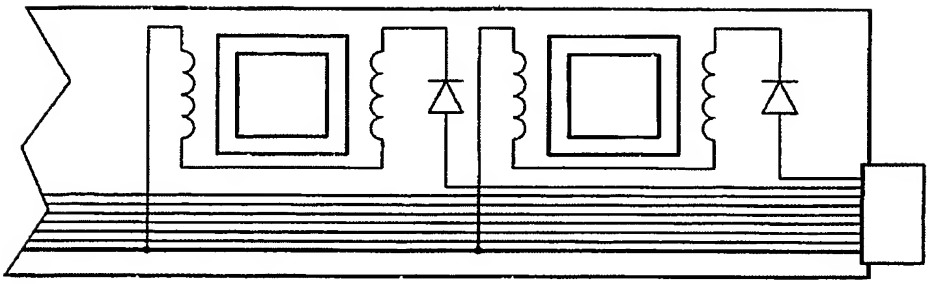


FIG.6D

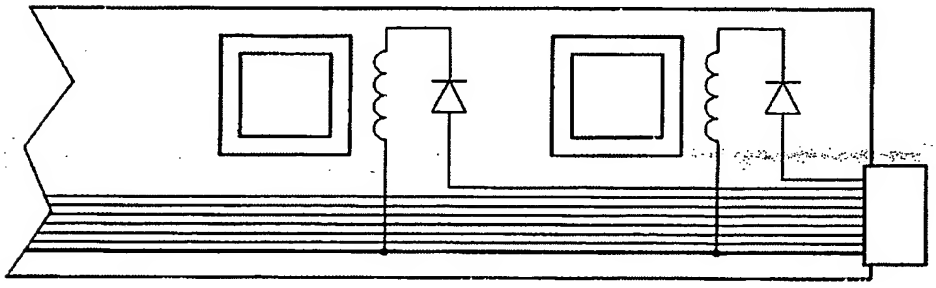


FIG.6E

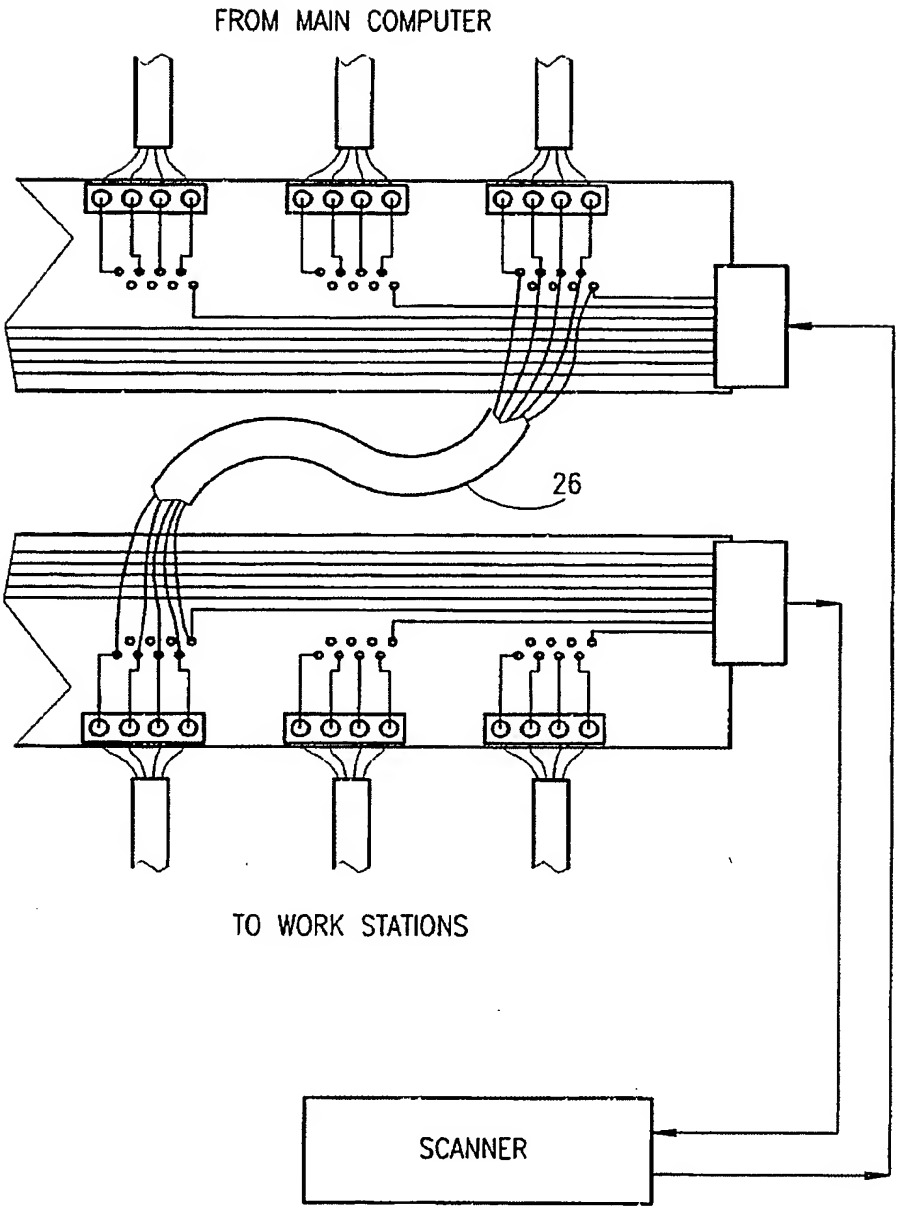


FIG.7

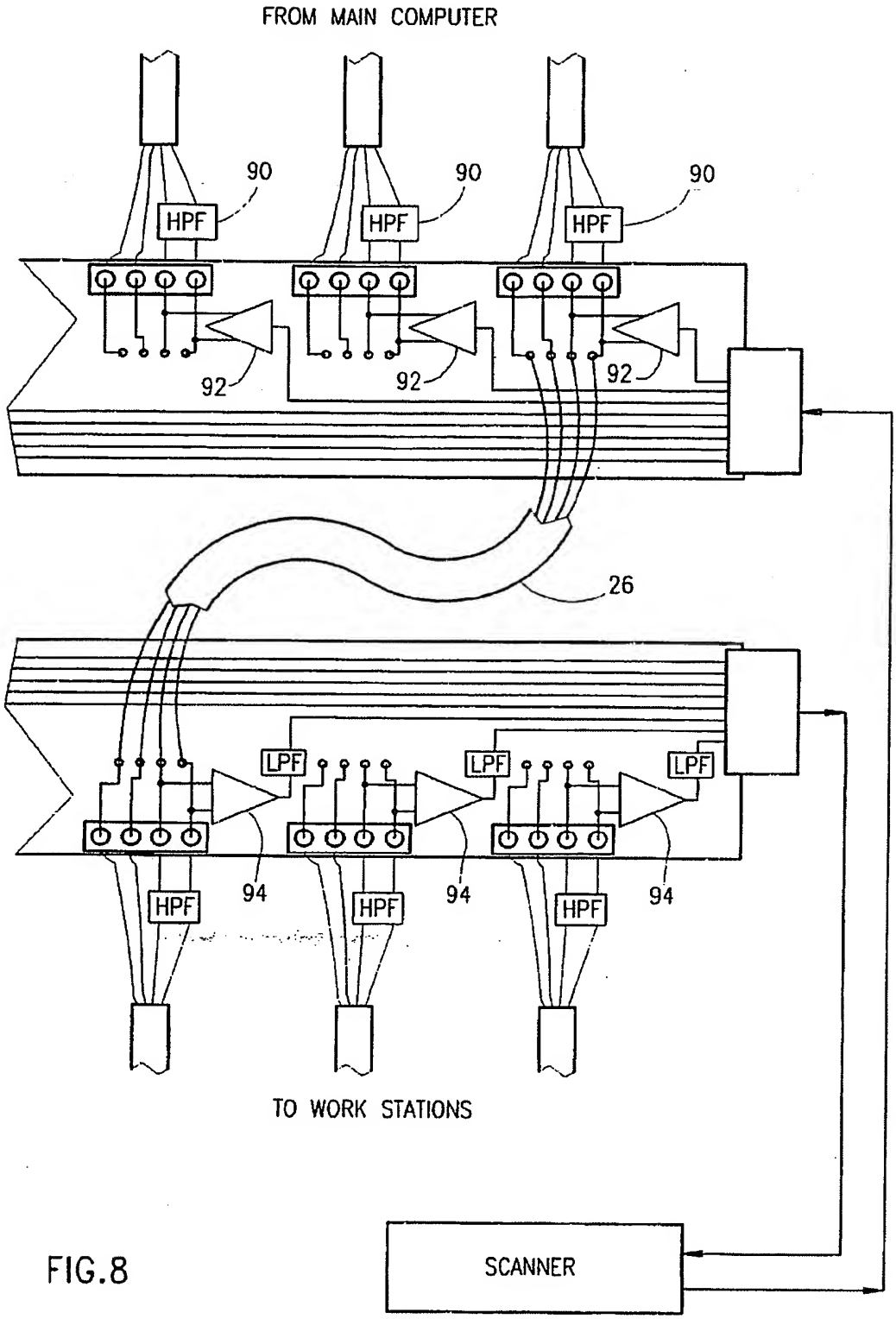

FROM MAIN COMPUTER
90
90
90
HPF
HPF
HPF
92
92
92
26
LPF
LPF
LPF
94
94
94
HPF
HPF
HPF
TO WORK STATIONS
SCANNER

FIG.8

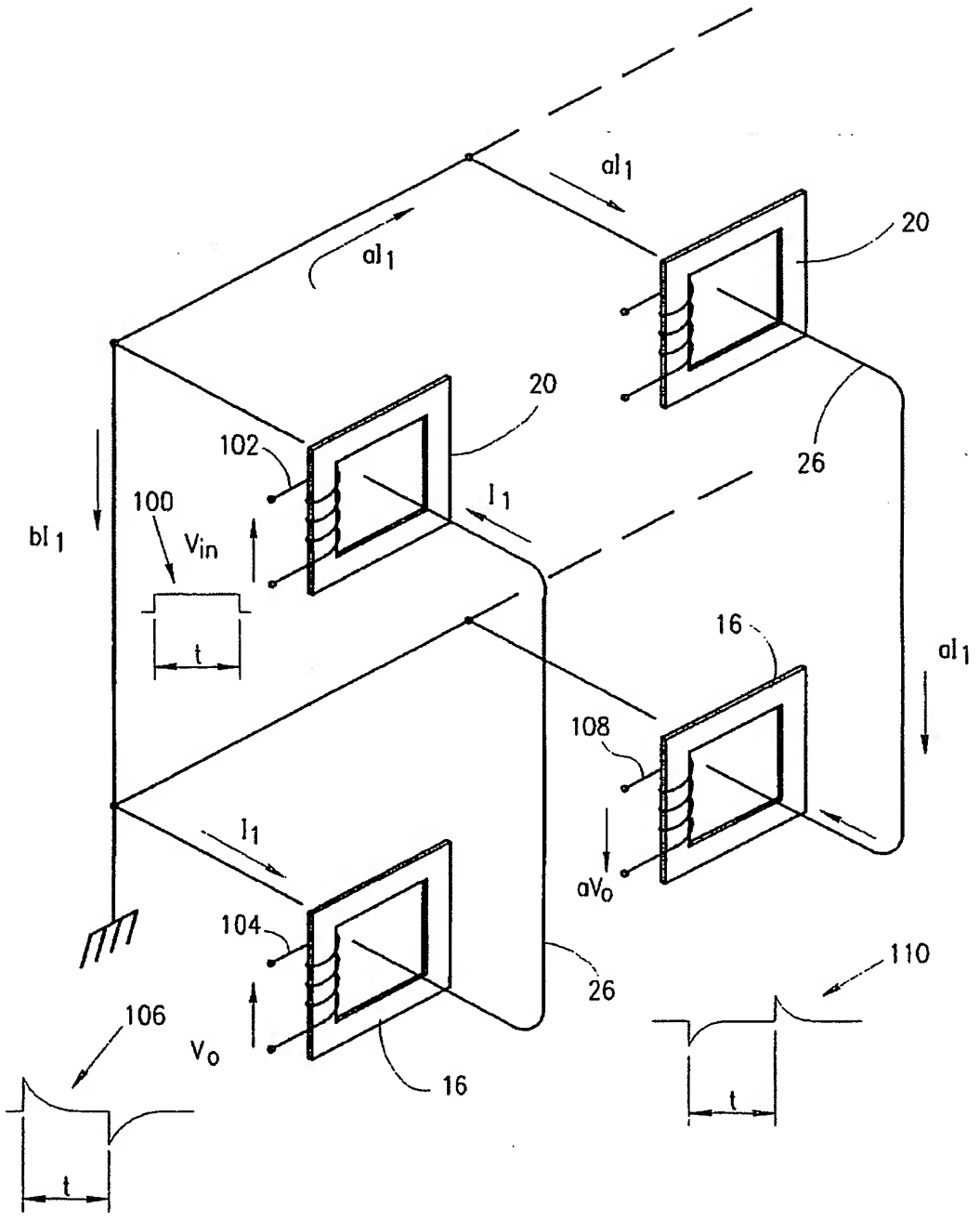

aI1
aI1
20
20
102
26
100
I1
bI1
Vin
t
16
aI1
108
I1
aVo
104
110
26
106
Vo
16
t
t

FIG.9

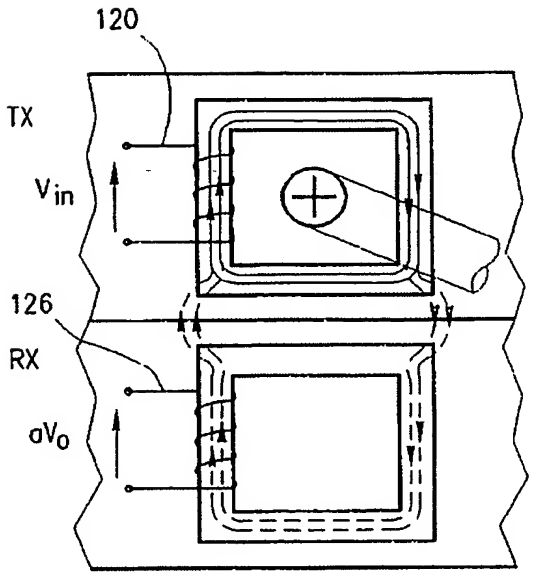


FIG.10A

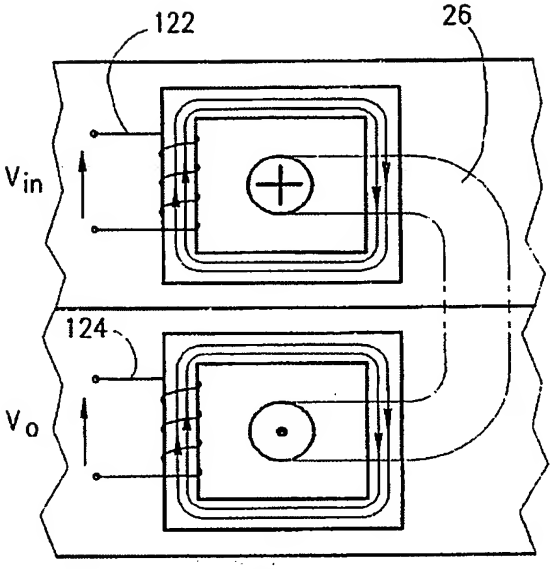
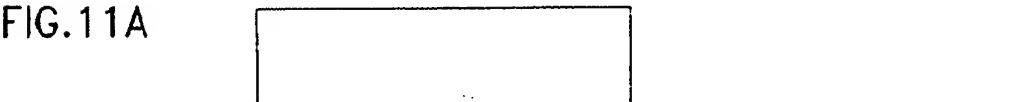


FIG.10B

FIG.11A



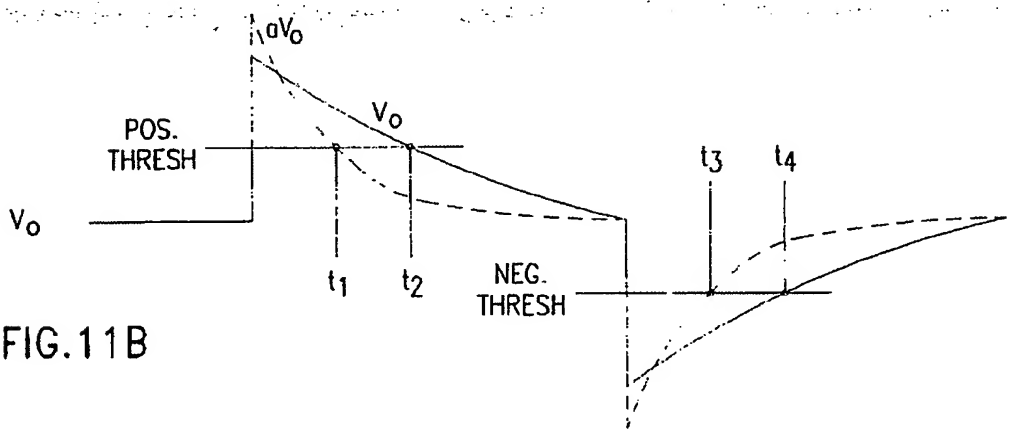


FIG.11B

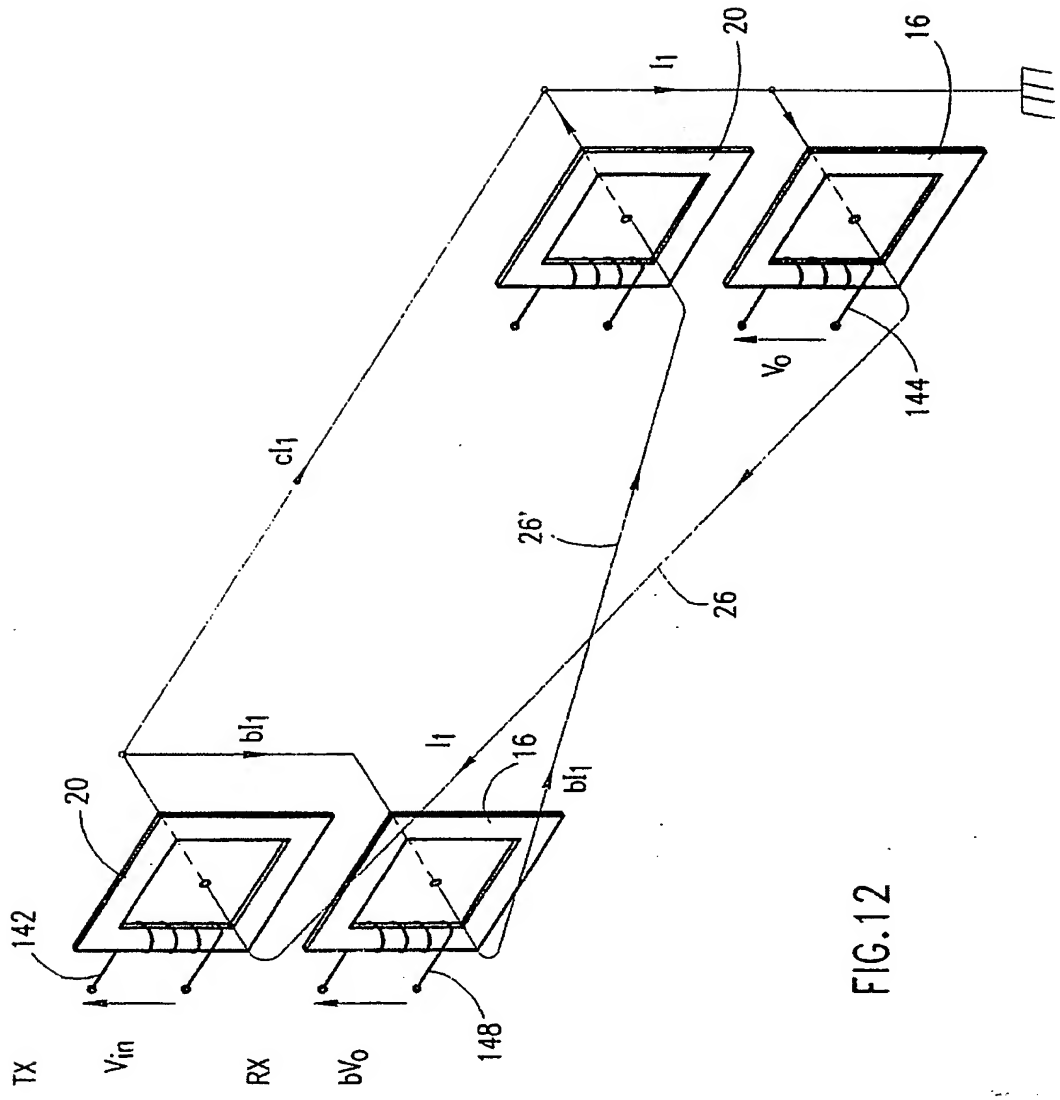


FIG.12

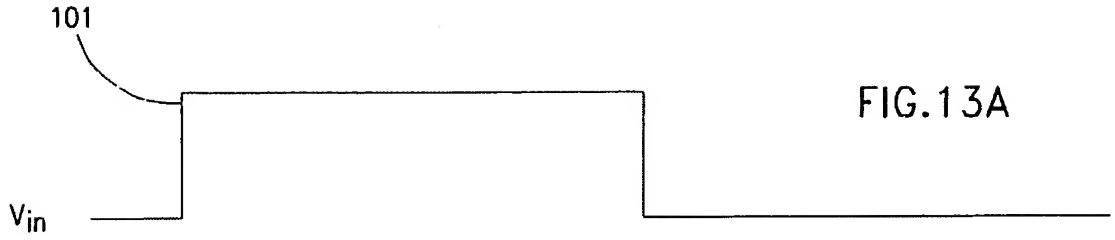


FIG.13A

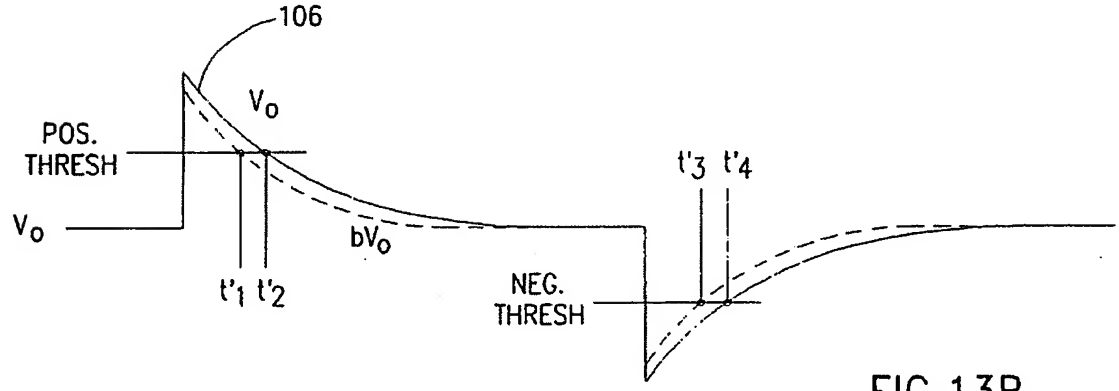


FIG.13B

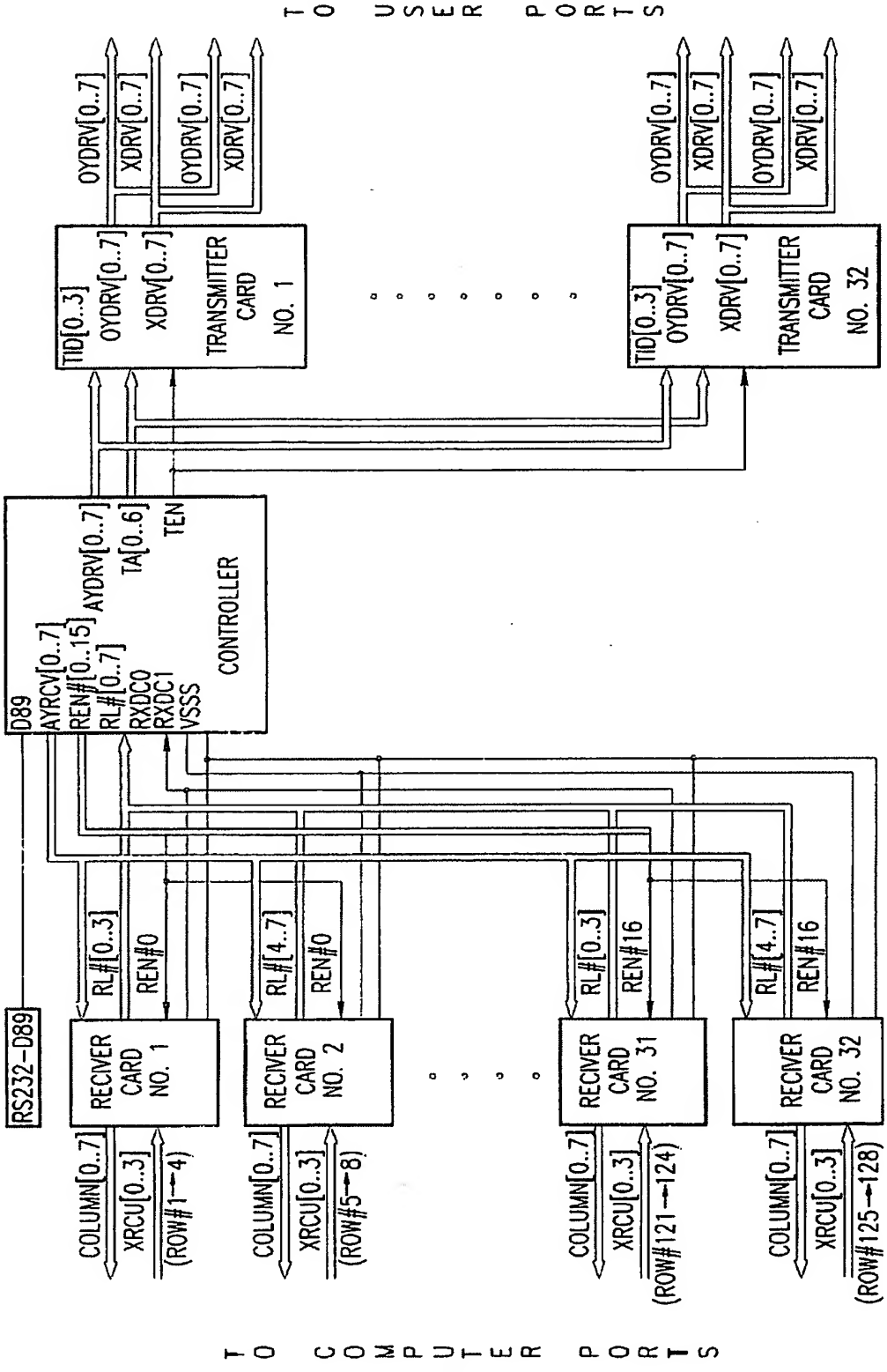

TO USER PORTS
OYDRV[0..7]
XDRV[0..7]
TID[0..3]
TRANSMITTER CARD NO. 1
TRANSMITTER CARD NO. 32
D89
AYRCV[0..7]
REN#[0..15]
RL#[0..7]
RXDC0
RXDC1
VSSS
AYDRV[0..7]
TA[0..6]
TEN
CONTROLLER
RS232-D89
RL#[0..3]
REN#0
REN#16
RL#[4..7]
RECIVER CARD NO. 1
RECIVER CARD NO. 2
RECIVER CARD NO. 31
RECIVER CARD NO. 32
COLUMN[0..7]
XRCU[0..3]
(ROW#1-4)
(ROW#5-8)
(ROW#121-124)
(ROW#125-128)
TO COMPUTER PORTS

FIG.14

FIG.15A

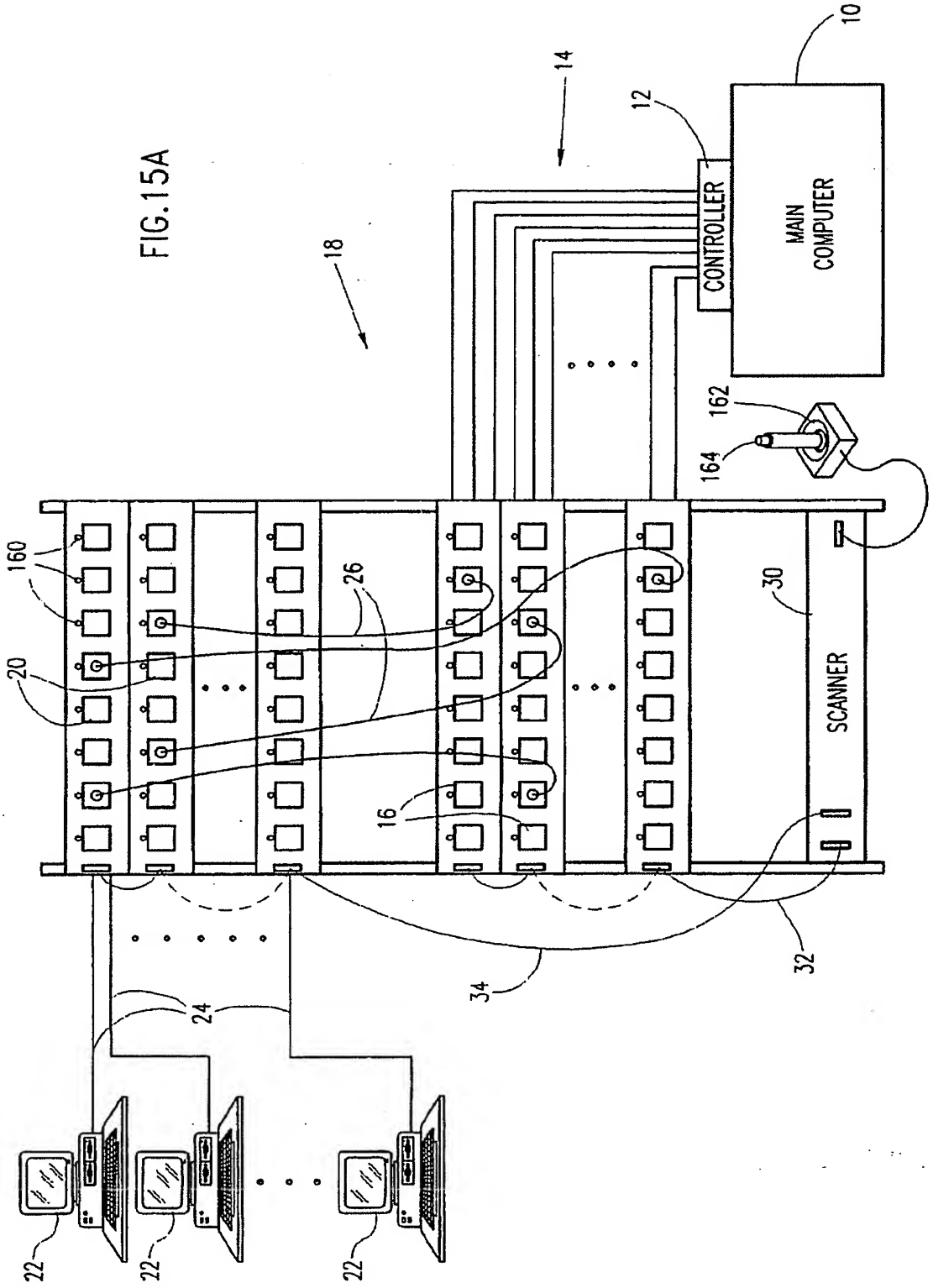
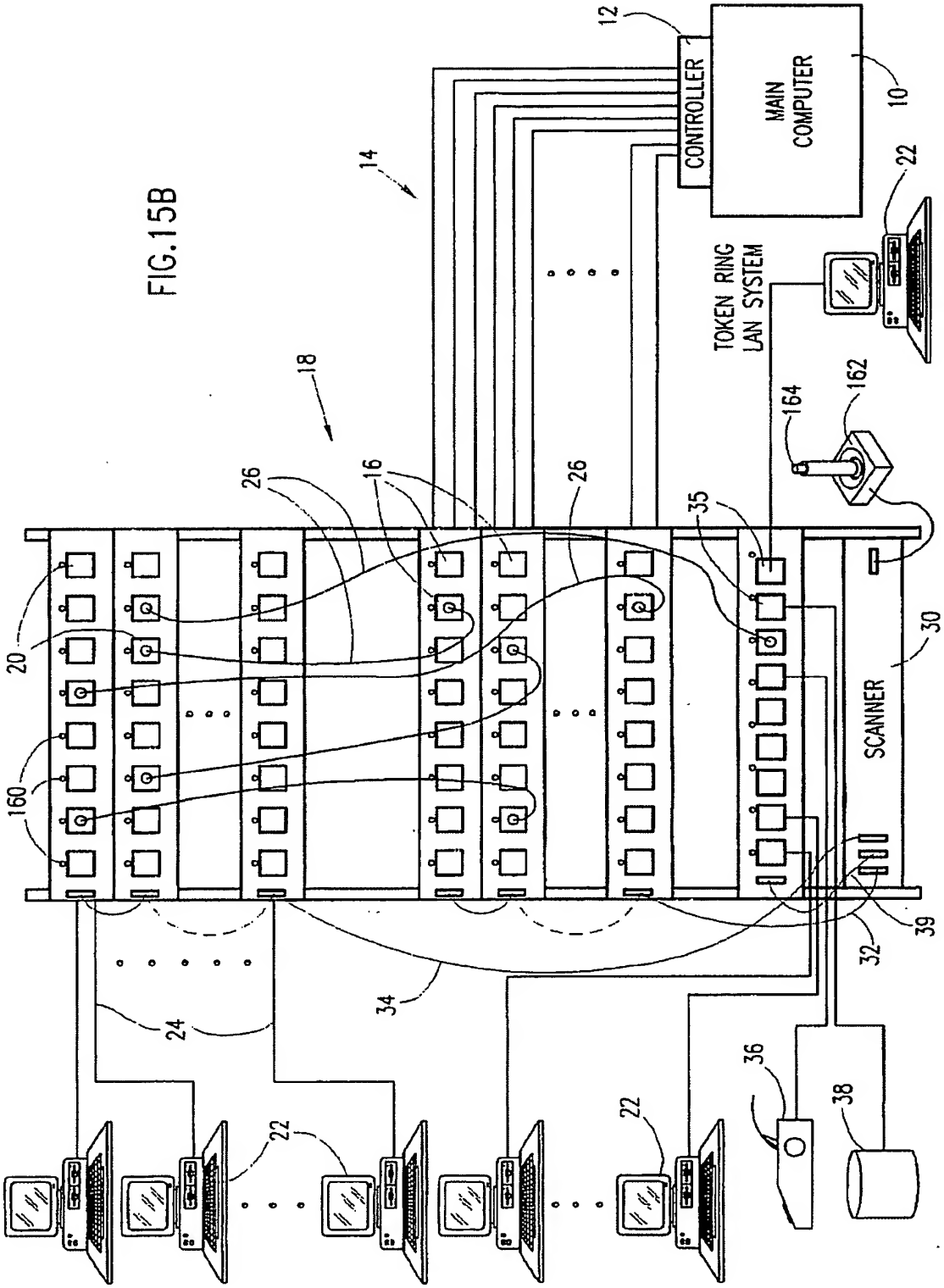

10
14
12
MAIN COMPUTER
CONTROLLER
18
162
164
160
20
26
30
SCANNER
16
32
34
24
22
22
22

FIG.15B


12
CONTROLLER
MAIN COMPUTER
10
14
22
TOKEN RING LAN SYSTEM
18
164
162
26
16
26
35
20
30
SCANNER
160
39
32
34
24
36
38
22
22

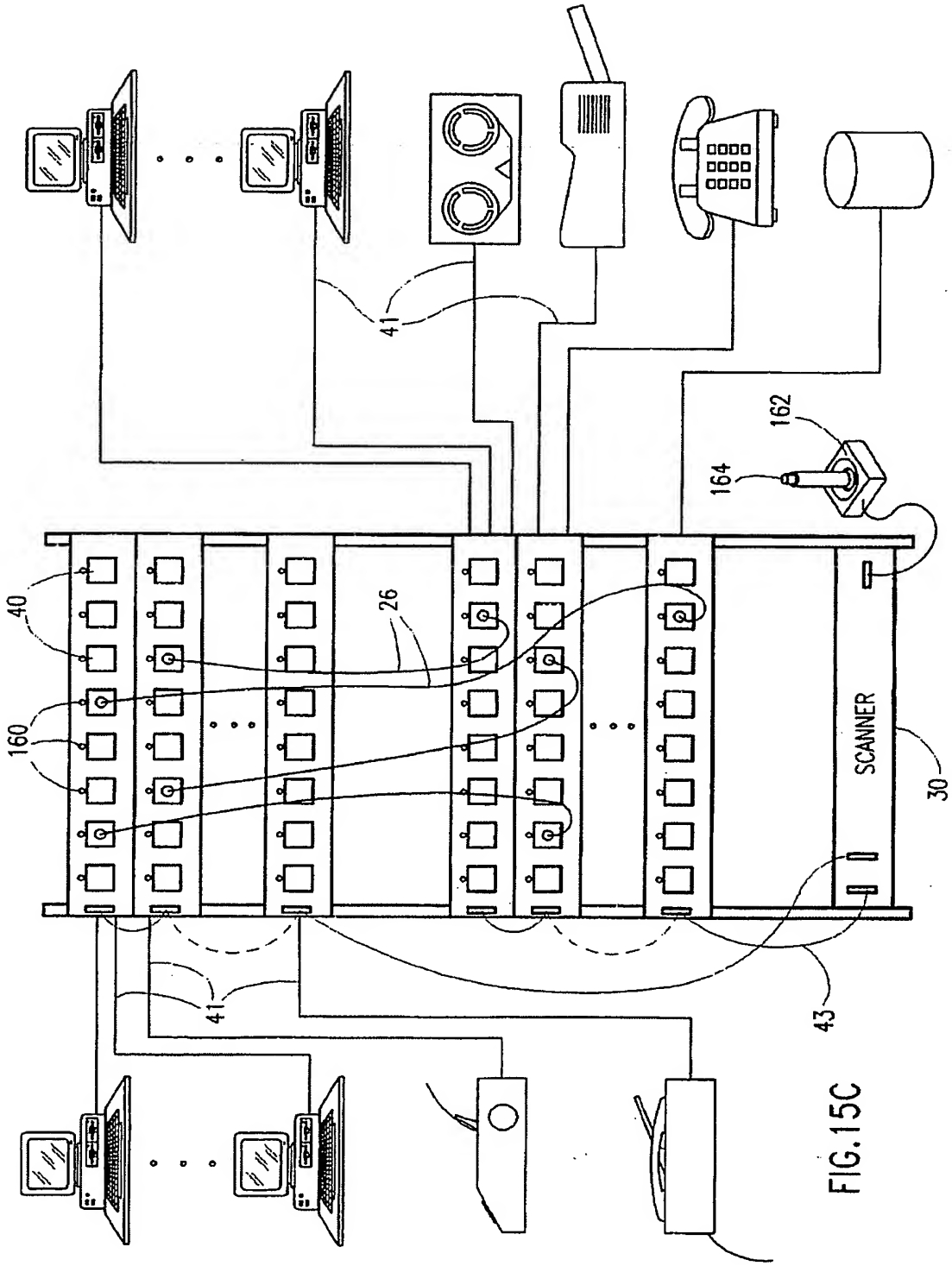


FIG.15C

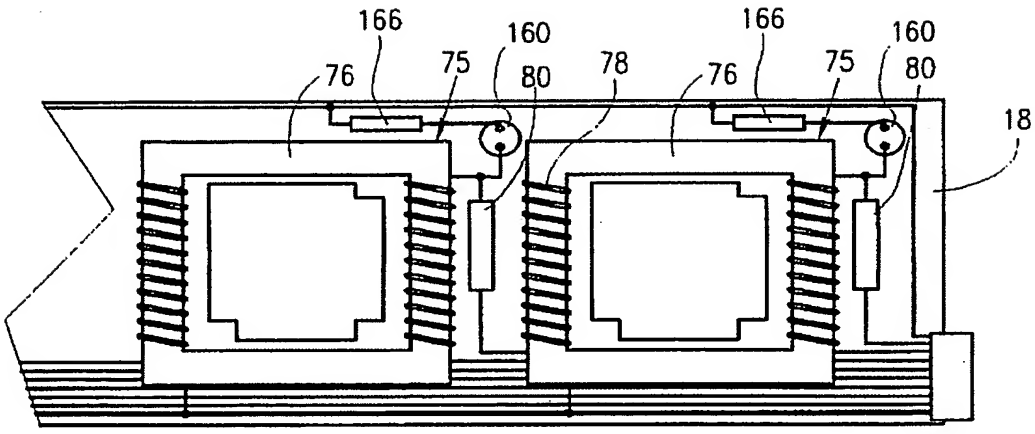


FIG.16

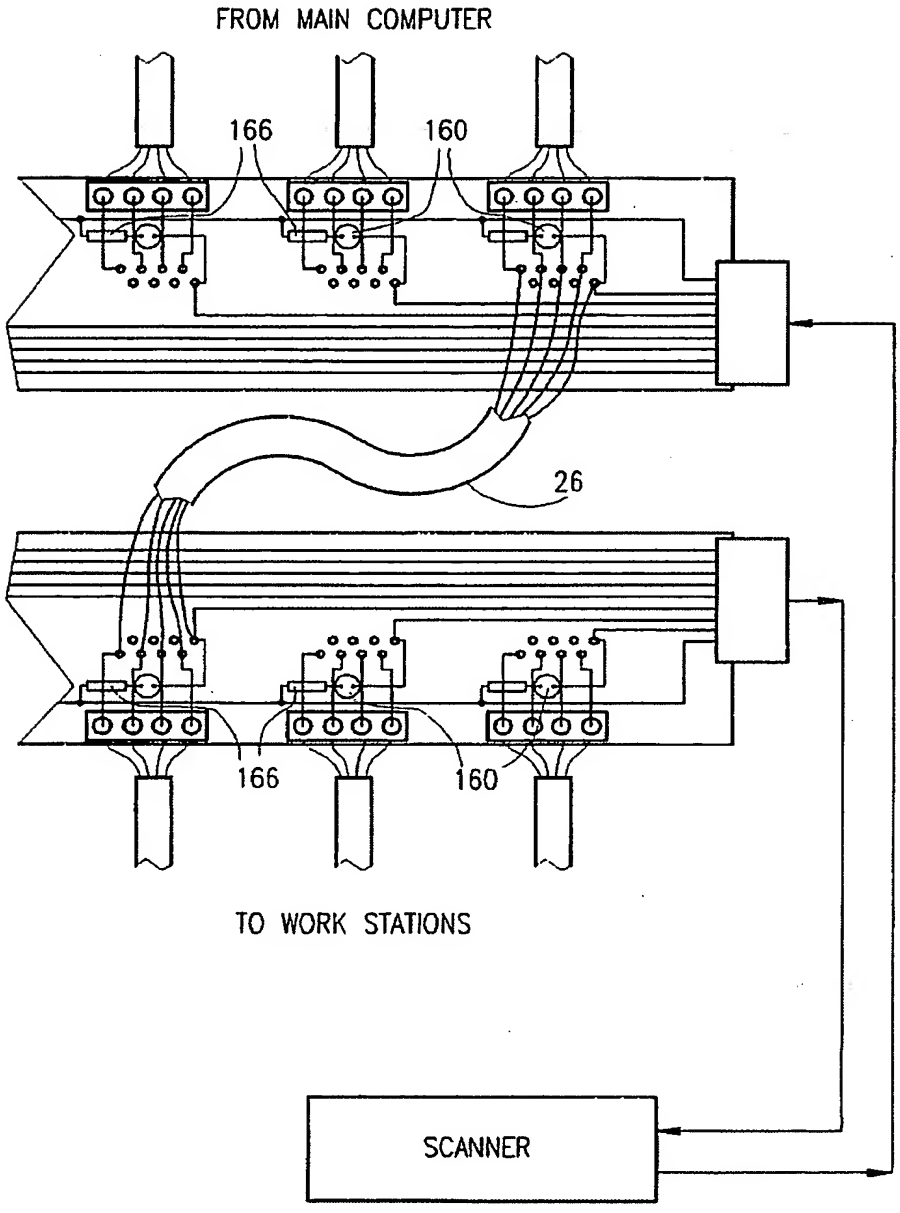

FROM MAIN COMPUTER
166
160
26
166
160
TO WORK STATIONS
SCANNER

FIG.17

European Patent Office

EUROPEAN SEARCH REPORT

Application Number
EP 93 30 4514

DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (Int. Cl.5)
X	GB-A-2 236 398 (CARTER) * abstract * * page 4, line 25 - page 5, line 25 *	1,5,7,9,14,15	G01R31/02 H04L12/26 H04Q1/14
Y		6,8	
A		2-4,13	
Y	AT-A-357 634 (GERASSIMOS) * page 3, line 5 - line 8 * * page 3, line 35 - line 40 *	6,8	
P,X	FR-A-2 680 067 (ALCATEL) * the whole document *	1,5,7,9-15	

TECHNICAL FIELDS SEARCHED (Int. Cl.5)

H04L
H04M
G01R

The present search report has been drawn up for all claims

Place of search	Date of completion of the search	Examiner
THE HAGUE	22 SEPTEMBER 1993	MIKKELSEN C.

CATEGORY OF CITED DOCUMENTS

X : particularly relevant if taken alone
Y : particularly relevant if combined with another document of the same category
A : technological background
O : non-written disclosure
P : intermediate document

T : theory or principle underlying the invention
E : earlier patent document, but published on, or after the filing date
D : document cited in the application
L : document cited for other reasons

& : member of the same patent family, corresponding document

EPO FORM 1503 03.82 (P0401)